PHILIPS



CURSUS BEDRIJFSELEKTRONICA

Inleiding tot de microcomputer

Leerlingboek ES

VOORLOPIGE UITGAVE

Philips Nederland B.V. - Afd. Onderwijsactiviteiten

© N.V. Philips' Gloeilampenfabrieken, Eindhoven, Nederland 1980

*Alle rechten uitdrukkelijk voorbehouden.
Vermenigvuldiging of mededeling aan derden,
in welke vorm ook, is zonder schriftelijke
toestemming van eigenares niet geoorloofd.*

PHILIPS



CURSUS BEDRIJFSELEKTRONICA

Inleiding tot de microcomputer

Leerlingboek ES

VOORLOPIGE UITGAVE

Philips Nederland B.V. - Afd. Onderwijsactiviteiten

OVER DEZE SCANS
Als basis voor deze scans hebben wij gebruik gemaakt van de door 'Freeservicemanuals' in 2018 gemaakte scans. Wij hebben de pagina's van deze scans echter zorgvuldig naar de originele staat gerestaureerd, onder andere door alle persoonlijke notities en de antwoorden op alle oefeningen en vragen te verwijderen.

© N.V. Philips' Gloeilampenfabrieken, Eindhoven, Nederland 1980

*Alle rechten uitdrukkelijk voorbehouden.
Vermenigvuldiging of mededeling aan derden,
in welke vorm ook, is zonder schriftelijke
toestemming van eigenares niet geoorloofd.*

INHOUDSOPGAVE

1

LOGICASYMBOLEN VOOR ONTWERP- EN UITVOERINGSSCHEMA'S

INLEIDING

Op het gebied van binaire logica bestaat tot nu toe nogal verwarring over grafische symbolen die logische functies moeten weergeven. Er zijn verschillende systemen in gebruik en het is soms zeer lastig van het ene systeem naar het andere over te schakelen. Internationale instanties hebben één systeem ontworpen dat aan deze verwarrende situatie een einde moet maken. Aangezien dit systeem internationaal is goedgekeurd en grote fabricanten van IC's het op korte termijn zullen invoeren, is deze cursus reeds aangepast aan de nieuwe normen.

De bedoeling van dit hoofdstuk is de basisbegrippen van de internationale normen duidelijk te maken.

SOORTEN SCHEMA'S

In binaire logica kennen we twee soorten schema's waarin symbolen voor logische functies en voor logica-elementen worden toegepast, namelijk:

- het ontwerpschema,
- het uitvoeringsschema.

Het ontwerpschema is een theoretisch logicaschema dat het functionele gedrag van een binair systeem vastlegt, waarbij nog geen rekening wordt gehouden met de diverse realisatiemogelijkheden. Het uitvoeringsschema verstrekt de technische uitvoering van een binair systeem.

Het onderscheiden van deze twee soorten schema's, ieder met zijn eigen symbolen en afspraken, is voortgekomen uit de omstandigheid dat een binair logisch systeem op verschillende manieren gerealiseerd kan worden. De automatische besturing van een bepaalde machine kan b.v. met pneumatische onderdelen uitgevoerd worden. Hij kan echter ook worden uitgevoerd met elektronische of met mechanische onderdelen.
Het ontwerpschema onderscheidt nog niet welke techniek wordt gebruikt.
De te gebruiken techniek bepaalt hoe het uitvoeringsschema er zal uitzien.

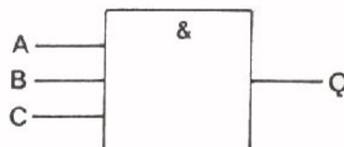
SAMENSTELLING VAN HET BOUWSTEENSYMBOOL

Zowel in ontwerpschema's als in uitvoeringsschema's bestaat het bouwsteensymbool uit een kader (= een rechthoek in de voorbeelden van deze les) of een combinatie van kaders met één of meer functiesymbolen. U kent waarschijnlijk reeds enkele functiesymbolen, zoals:

& voor een AND-functie

≥1 voor een OR-functie

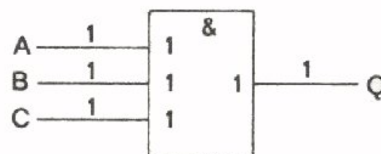
Het bouwsteensymbool bevat meestal één of meer ingangen en één of meer uitgangen. De ingangen worden doorgaans aan de linkerkant van het symbool geplaatst en de uitgangen aan de rechterkant. Het functiesymbool staat, bij voorkeur, boven en in het midden binnen het kader. Hieronder ziet U het symbool van een AND-poort met drie ingangen A, B en C en een uitgang Q.



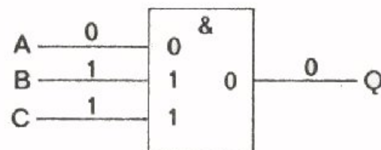
De lengte - breedteverhouding van de symboolkaders is vrij te kiezen. Er zijn nog meer algemene regels voor de samenstelling van de bouwsteensymbolen. Deze komen later ter sprake wanneer we ze nodig hebben.

HET ONTWERPSCHEMA SYMBOOL

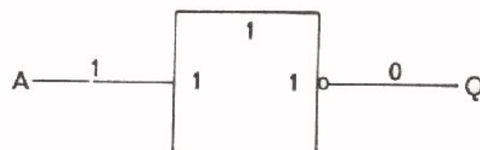
In schema's voor systeemontwerp wordt het functionele gedrag van een binair systeem beschreven. De twee logische toestanden "waar" en "niet waar" van een logisch signaal worden voorgesteld door de symbolen "1" en "0". Voor een AND-poort bijvoorbeeld is de uitgang alleen "waar" als alle ingangen "waar" zijn. Deze situatie kunnen we beschrijven met de volgende tekening.



In alle andere gevallen, is de uitgang "niet waar". Dus, bijvoorbeeld:

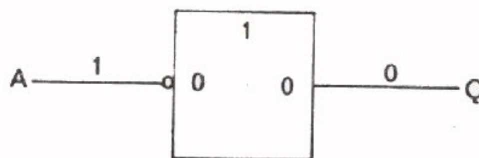


De toevoegingen 0 en 1 maken geen deel uit van het bouwsteensymbool. Ze worden gebruikt om de logische toestanden weer te geven. We onderscheiden de externe logische toestand (de toestand buiten het kader) en de interne logische toestand (de toestand binnen het kader). In de gevallen hierboven is te zien dat de interne logische toestand en de externe logische toestand voor elke ingang en voor de uitgang altijd gelijk zijn. Dit laatste is niet het geval voor het symbool van een NOT-element. Bekijk de situatie hieronder.



De externe logische toestand aan de ingang is 1, de interne logische toestand van de ingang is ook 1. Het functiesymbool 1 geeft aan dat de logische toestand van de uitgang van het element binnen het kader gelijk is aan de logische toestand van de ingang binnen het kader. De externe logische toestand van de uitgang is echter niet gelijk aan zijn interne logische toestand. Dit wordt aangegeven met het cirkeltje aan de uitgang. Dit cirkeltje is een z.g. negatie-indicator, die aanduidt dat 0 in 1, of 1 in 0 overgaat. Men zegt: de negatie-indicator "negeert".

Men kan het NOT-element ook als volgt weergeven:



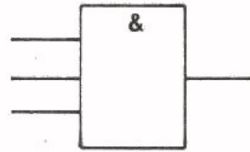
Ga dit voor Uzelf na !

FUNDAMENTELE BOUWSTENEN IN ONTWERPSCHEMA'S

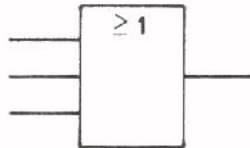
Voor het ontwerpen van een binair systeem beschikt een ontwerper over AND-poorten, OR-poorten en NOT-elementen. De te verwerken informatie wordt met 1 (= "waar") en 0 (= "niet waar") aangeduid. Worden aan een systeem b.v. de signalen A, B en C toegevoerd en wordt het signaal Q afgenomen, dan kan men in een z.g. waarheidstabel weergeven welke combinaties van A, B en C (uitgedrukt in 0, resp. 1) niet en welke welke wel een uitgangs-signaal Q = 1 tot gevolg hebben.

De fundamentele ontwerpschema - symbolen zijn:

AND – poort



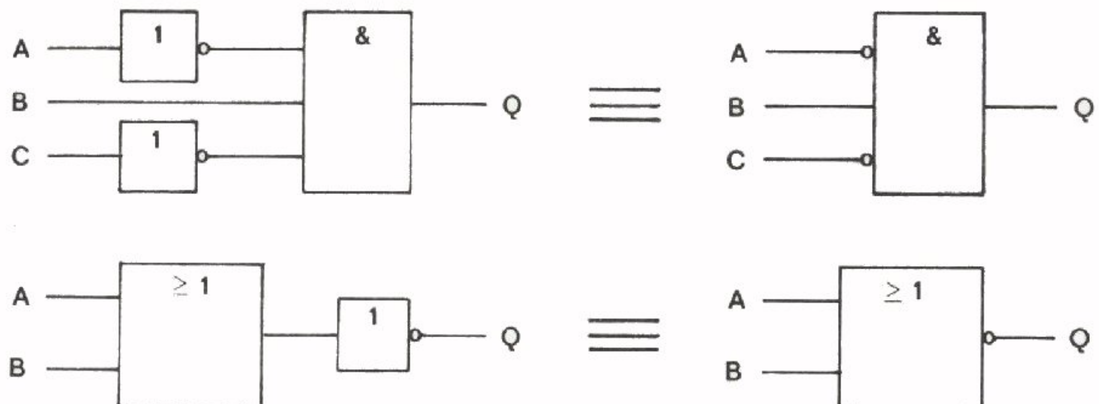
OR–poort



NOT–element



Het is handig en aan te bevelen om in een ontwerpschema niet de NOT-elementsymbolen te gebruiken, maar in plaats daarvan de NOT-functie als cirkeltje aan de ingang (resp. uitgang) van de aangrenzende poort weer te geven. Zie voorbeelden hieronder:



Het ontwerpschema is een theoretisch schema, waarin nog helemaal niet tot uitdrukking komt welke techniek (hydraulisch, mechanisch, elektrisch, enz.) men gaat gebruiken om het systeem te realiseren. Er is ook nog geen sprake van welke digitale bouwstenen we in een bepaalde techniek zullen gebruiken. De keuze van de toe te passen techniek en bouwstenen wordt naderhand gedaan; in geval van een elektrische realisatie kan men nog kiezen uit TTL, CMOS, ECL, enz. Hierbij kunnen o.a. economische overwegingen een belangrijke rol spelen. Hebben we de keuze vastgelegd, dan moet het ontwerpschema omgezet worden in het uitvoeringsschema. Dit gaan we later in dit hoofdstuk bekijken.

HET ONTWERPSCHEMA VAN EEN ROLFILM-AUTOMAAT

We beginnen een voorbeeld te behandelen van het opzetten van een ontwerpschema. Dit voorbeeld zal verderop telkens weer ter sprake komen.
Bij een rolfilm-automaat moet na het inwerpen van de benodigde munten en het drukken op één van de drie aanwezige keuzeknoppen eerst een "teller van het aantal afgegeven rolfilms" een stap verdergaan; vervolgens moet men óf een rolfilm A, óf een rolfilm B, óf een rolfilm C kunnen verkrijgen.
De automaat moet geen film afgeven als tegelijkertijd op 2 of 3 keuzeknoppen wordt gedrukt. Men wil dit bereiken met behulp van een digitaal systeem.
Drukken op de A-knop heeft tot gevolg dat A = 1 wordt. Drukken op de B-knop (resp. de C-knop) heeft tot gevolg dat B = 1 (resp. C = 1) wordt.
Wordt alléén A = 1 (of B = 1 of C = 1), dan moet F = 1 worden waardoor de teller een stap verdergaat. Als alléén A = 1 (of B = 1 of C = 1) wordt, dan moet $F_A = 1$ (of $F_B = 1$ of $F_C = 1$) worden en kan het schuiflaadje met de gewenste rolfilm daardoor worden opengetrokken.

A	B	C	F	F_A	F_B	F_C
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0

De waarheidstabel van het vereiste digitale systeem is hiernaast weergegeven. Slechts als één van de signalen A, B of C gelijk is aan 1, is ook F = 1 en telt de teller een verder.

Is b.v. A = 1, dan moeten F = 1 en A = 1 tesamen voor een resultaat $F_A = 1$ zorgen, dat mogelijk maakt dat alleen het A-laadje opengetrokken kan worden. Uit de waarheidstabel kunnen we afleiden:

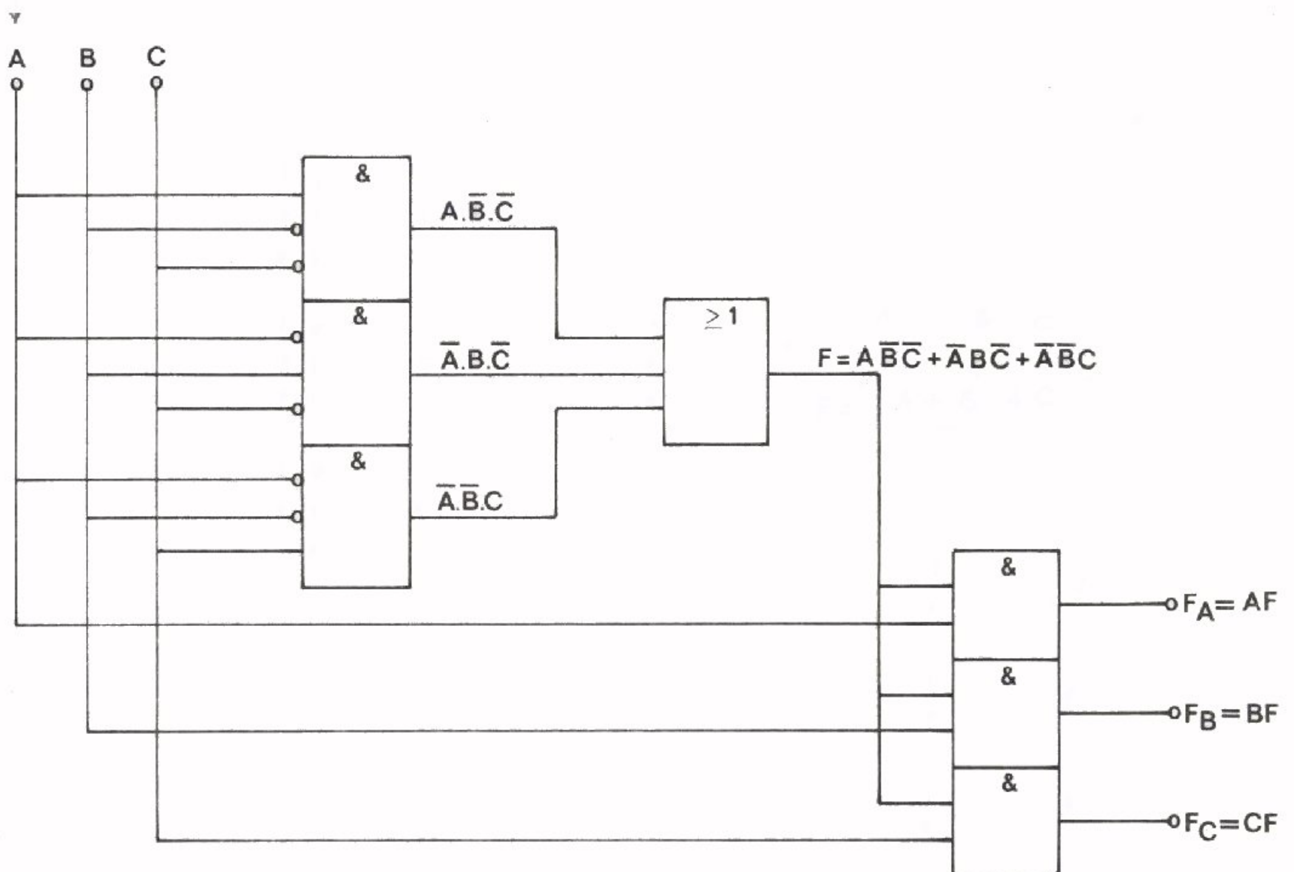
$$F = A.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + \overline{A}.\overline{B}.C$$

$$F_A = A.F$$

$$F_B = B.F$$

$$F_C = C.F$$

Aan de hand van deze vergelijkingen valt het volgende ontwerpschema op te stellen.



In de bovenstaande tekening is bij de AND-poorten een nieuwe regel voor het tekenen van symbolen toegepast. Het is namelijk toegestaan (dus niet verplicht) om bouwsteensymbolen op deze manier tegen elkaar aan te tekenen, mits er tussen de elementen geen logische verbinding is; d.w.z., de elementen moeten onafhankelijk van elkaar zijn. Er zijn nog meer mogelijke combinaties van kaders bedacht; die zult U later ontmoeten. Deze regels zijn ingevoerd om ruimte te sparen.

OPMERKING

Het rolfilmautomatenschema kan ook eenvoudiger gehouden worden. We hebben het gecompliceerder dan nodig gemaakt om later de omzetting van een ontwerpschema in een uitvoeringsschema beter te kunnen demonstreren.

HET SYSTEEM VAN DE ONTWERPSCHEMA - SYMBOLEN

We geven nu een samenvatting van de eigenschappen en het gebruik van ontwerpschema - symbolen.

Ontwerpschema - symbolen zijn bedoeld om gebruikt te worden bij het ontwerpen van digitale schakelingen zolang de latere praktische uitvoering nog niet in de beschouwingen wordt betrokken.

Met de 3 basisfuncties AND, OR en NOT kan men elke digitale schakeling bouwen. Het verdient daarbij aanbeveling om NOT-elementen niet apart te tekenen, maar ze als negatie-indicatoren aan ingangen of uitgangen van aangrenzende digitale elementen te zetten.

Bij ontwerpschema-symbolen houdt men zich aan de volgende afspraken:

- De NOT-functie wordt altijd BUITEN het kader van een schemasymbool gehouden en wordt vertegenwoordigd door een NEGATIE-INDICATOR.
- De overige functies (zoals & en ≤ 1) worden altijd BINNEN het kader gehouden.
- Zowel binnen als buiten de kaders worden de informatiewaarden aangeduid met 0 (= niet waar) en 1 (= waar).
- Een negatie-indicator "negeert"; d.w.z., hij zet 0 om in 1 en omgekeerd 1 om in 0.

Het gedrag van een digitale schakeling is aan de hand van het ontwerpschema vast te leggen in een waarheidstabel en/of volgordetabel, waarin 0 en 1 worden ingevuld:

- Elke regel van een WAARHEIDSTABEL is op zichzelf altijd geldig, onverschillig wat er aan de vermelde toestand is voorafgegaan.
- Een regel van een VOLGORDETABEL behoeft NIET op zichzelf altijd waar te zijn omdat de voorafgaande toestand soms van invloed is. Zodra een digitale schakeling een of meer geheugenelementen bevat, heeft de voorgaande toestand soms invloed op de volgende.

HET UITVOERINGSSCHEMA

Heeft men een technisch probleem (zoals dat van de eerder ter sprake gekomen rolfilmautomaat) theoretisch opgelost in de vorm van een ontwerpschema, dan dient men daarna tot een praktische uitvoering te komen. Eerst moeten de te gebruiken digitale bouwstenen gekozen worden, waarna men tot het uitvoeringsschema moet komen waarin van deze bouwstenen gebruik wordt gemaakt.

De bouwstenen kunnen hydraulische, pneumatische, elektrische of andere bouwstenen zijn. Wij beperken ons in deze lessen tot ELEKTRISCHE DIGITALE BOUWSTENEN in TTL-uitvoering.

In het (theoretische) ONTWERPSCHEMA is steeds sprake van twee mogelijke signalen: 0 en 1 of "niet waar" en "waar".

In het (praktische) UITVOERINGSSCHEMA met elektrische bouwstenen heeft men buiten genoemde bouwstenen te doen met twee mogelijke signalen in de vorm van ELEKTRISCHE SPANNING. De ene spanningswaarde die kan optreden wordt aangeduid met H; de andere, minder positieve, met L. In deze lessen gebruiken we T T L - geïntegreerde schakelingen, waarbij

H (= hoog niveau) overeenkomt met ≃ 3,5 V.

en L (= laag niveau) overeenkomt met ≃ 0 V.

HET SYSTEEM VAN DE UITVOERINGSSCHEMA-SYMBOLEN

- De uitvoeringsschema-symbolen zijn in principe gelijk aan de ontwerpschema-symbolen. Er zijn echter twee verschillen:

  1. BINNEN de kaders worden de signaalwaarden altijd aangeduid met 0 (= niet waar) en 1 (= waar). BUITEN de kaders worden de signaalwaarden daarentegen aangeduid met L (= laag spanningsniveau) en H (= hoog spanningsniveau).

  2. Voor de NOT-functie wordt NIET de NEGATIE-INDICATOR (het cirkeltje) gebruikt. Deze functie wordt anders weergegeven, hetgeen later wordt uiteengezet.

- Het verband tussen de spanningsniveau's H en L buiten het kader en de signaalwaarden 0 en 1 binnen het kader wordt als volgt vastgelegd:

  1. Tekent men een polariteitsindicator, dan gaat L buiten het kader samen met 1 binnen het kader en H buiten met 0 binnen.



ingangen uitgangen

  2. Tekent men géén polariteitsindicator, dan gaat H buiten het kader samen met 1 binnen het kader en L buiten met 0 binnen.



ingangen uitgangen

- Buiten de kaders (d.w.z. op de verbindingen) kunnen zich twee toestanden voordoen:

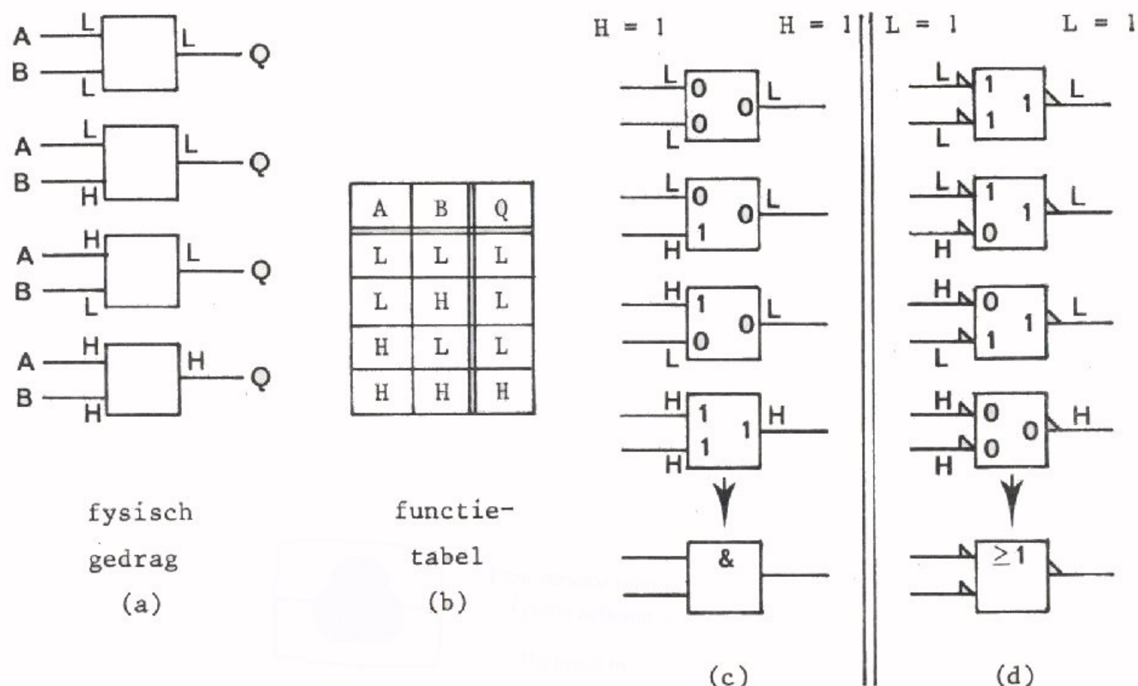
  óf voor het aanwezige signaal geldt H = 1 en L = 0,

  óf voor het aanwezige signaal geldt L = 1 en H = 0.

  We gaan nu verschillende uitvoeringsschema-symbolen behandelen.
  Het FYSISCHE gedrag van elke bouwsteen leggen we telkens overzichtelijk vast in een z.g. FUNCTIETABEL met de signaalwaarden L en H.
  Deze bij het uitvoeringsschema-symbool horende tabel komt overeen met de waarheidstabel die bij een ontwerpschema-symbool hoort.

DE ZOGENAAMDE AND-POORT

Het fysische gedrag van een elektrische digitale bouwsteen is na te gaan door diverse elektrische spanningen toe te voeren en te meten wat voor spanning er dan uit komt. De spanningen zijn daarbij óf laag (L), óf hoog (H). In ons eerste voorbeeld hebben we in fig. (a) het fysische gedrag weergegeven. Dit is in fig. (b) samengevat in een functietabel.



A	B	Q
L	L	L
L	H	L
H	L	L
H	H	H

fysisch gedrag (a)

functie- tabel (b)

(c)

(d)

Uit de functietabel zien we dat de uitgang alleen dan H is, als A en B beide H zijn. De functie kan dus worden weergegeven door een AND-poort, waarbij voor de uitgang en de beide ingangen H met 1 overeenkomt. Aan in- en uitgangen waar H met 1 overeenkomt, tekenen we geen polariteits-indicator. Het symbool wordt dus zoals onderaan in fig.(c) is weergegeven.

Uit de functietabel kunnen we ook aflezen dat de uitgang L is als een of meer ingangen L zijn. De functie kan dus ook worden weergegeven door een OR-poort, waarbij voor de uitgang en de beide ingangen L met 1 overeen-komt. Aan in- en uitgangen waar L met 1 overeenkomt, tekenen we een polariteitsindicator. Het symbool wordt dus zoals onderaan in fig. (d). is weergegeven.

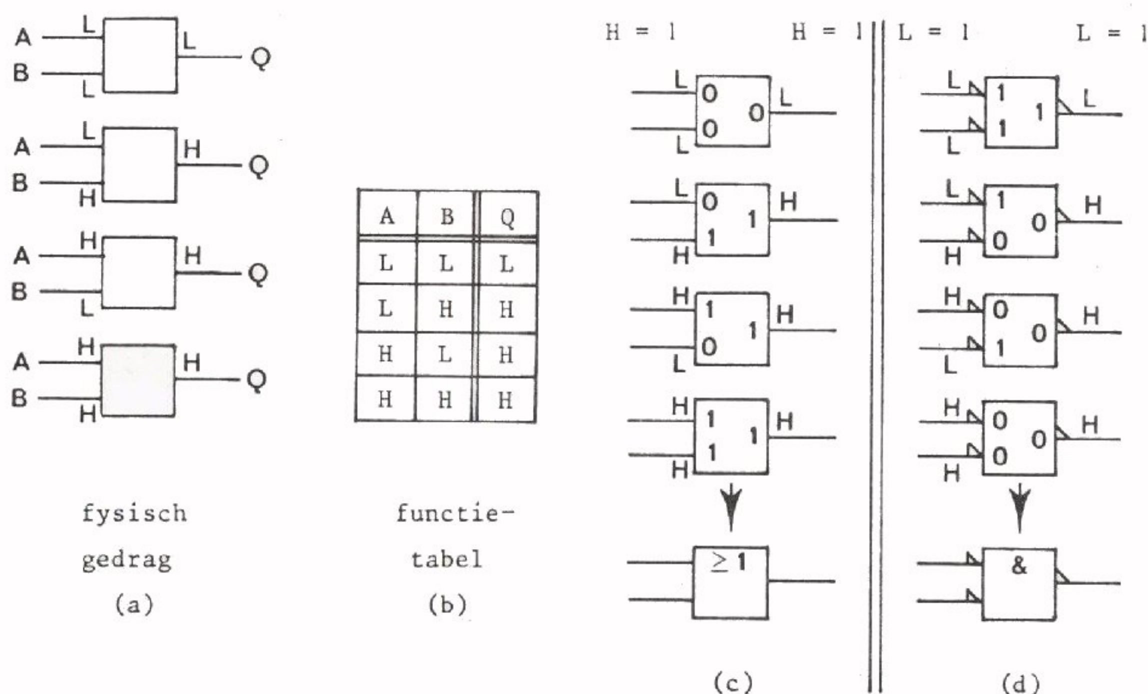
We zien dat de bouwsteen als AND-poort of als OR-poort kan dienen, afhankelijk daarvan of men H = 1 en L = 0 of L = 1 en H = 0 veronderstelt. Vroeger hield men meestal H = 1 en L = 0 aan en noemde men de bouwsteen daarom een AND-poort. Vandaar dat wij hier over de "zogenaamde" AND-poort hebben gesproken, hoewel hij net zo goed als OR-poort dienst kan doen.

We hebben gezien dat een zelfde fysische schakeling zowel de AND-functie als de OR-functie kan verrichten. Naar aanleiding daarvan merken we op dat de volgende ALGEMENE OMZETREGEL altijd blijkt op te gaan voor combi-natorische schakelingen (maar NIET voor sequentiële schakelingen zoals flip-flops, one-shots e.d.):

Men mag & vervangen door ≤ 1 (en omgekeerd), als men "de in- en uitgangen zonder polariteitsindicator" elk van een polariteitsindicator voorziet en omgekeerd van "de in- en uitgangen met polariteitsindicator" de polariteitsindicator weglaat.

DE ZOGENAAMDE OR-POORT

In het volgende voorbeeld is in fig. (a) weer het fysische gedrag en in fig. (b) de samenvattende functietabel gegeven.



A	B	Q
L	L	L
L	H	H
H	L	H
H	H	H

fysisch gedrag (a)

functie-tabel (b)

(c)

(d)

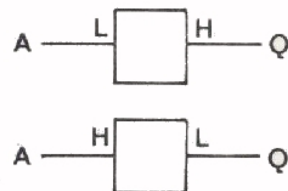
Uit de functietabel lezen we af dat Q = H als een of meer van de ingangen H is. Dit is de eigenschap van een OR-poort en het symbool is zoals onderaan in fig. (c) is weergegeven.

Uit de functietabel blijkt echter ook dat Q alleen L is als alle ingangen L zijn. Dit vertegenwoordigt een AND-functie, en het symbool daarvoor is zoals onderaan in fig. (d) is weergegeven.

Zoals U ziet gaat de omzetregel ook nu op.

DE INVERTER OF OMKEERTRAP

Het volgende voorbeeld is de inverter of omkeertrap. Voert men daaraan een lage spanning toe, dan is de uitgangsspanning hoog, en omgekeerd. In de onderstaande figuren (a) en (b) is dit weergegeven.



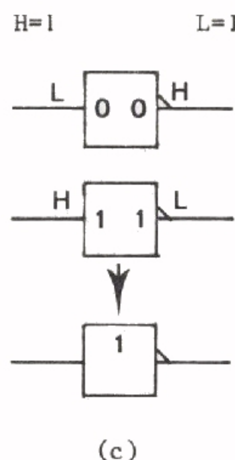
Fysische gedrag

(a)

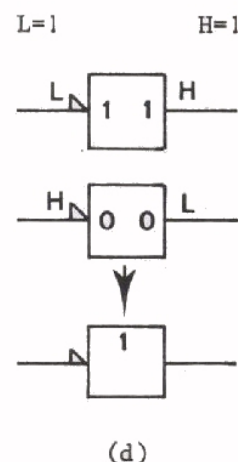
A	Q
L	H
H	L

Functie-tabel

(b)



(c)



(d)

Een NOT-functie binnen het kader vermijdend, zijn er weer 2 mogelijkheden om het element te zien:

- óf aan de ingang geldt H = 1 en L = 0, en aan de uitgang L = 1 en H = 0;
- óf aan de ingang geldt L = 1 en H = 0, en aan de uitgang H = 1 en L = 0.

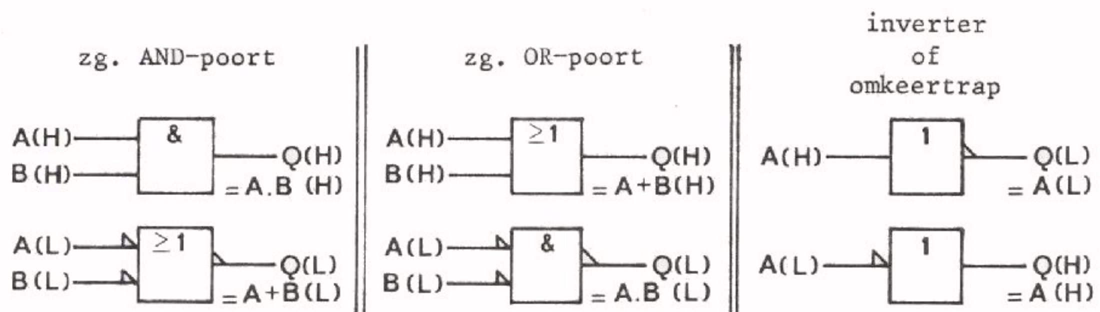
Dit kunt U zien in de figuren (c) en (d). Binnen het kader is de inverter een "doorverbinding". De polariteitsindicator buiten het kader geeft weer, dat men met behulp van een inverter:

- óf van de toestand waarbij H = 1 en L = 0 kan overgaan op die waarbij L = 1 en H = 0,
- óf van de toestand waarbij L = 1 en H = 0 kan overgaan op die waarbij H = 1 en L = 0.

Men kan een omkeertrap ook gebruiken om de NOT-functie te realiseren. Hierop komen we later in deze les terug.

HOE AAN TE DUIDEN DAT H=1 EN L=0, RESP. L=1 EN H=0

Een signaal wordt weergegeven met een letter (bijvoorbeeld de signalen A,B en Q in het AND-poort-voorbeeld) of met een Boole-formule. Aan deze letter of formule op zichzelf kam men niet zien of er geldt H=1 en L=0, hetzij L=1 en H=0.
Er is afgesproken om, als H=1 en L=0, achter de letter of formule (H) te zetten. Geldt daarentegen L=1 en H=0, dan zet men er (L) achter. We illustreren dit aan de hand van de drie tot nu toe gegeven voorbeelden. Deze zijn op de volgende bladzijde geschetst. Bestudeer dit eens goed.



Vaak legt men een signaalnaam zo vast, dat daaruit blijkt wanneer het signaal "waar" is, d.w.z. met 1 overeenkomst.

Zo kan men een alarmsignaal b.v. vastleggen met ALARM (H). Dit wil dan zeggen dat ALARM aanwezig is ("waar" is), als de signaalspanning hoog is.

Het is dan echter ook "waar", dat GEEN ALARM aanwezig is als de signaalspanning laag is. Dit is vast te leggen met $\overline{\text{ALARM}}$ (L).

Het besproken alarmsignaal valt dus te noteren:

- óf als ALARM (H), waarbij geldt H = 1 en L = 0;
- óf als $\overline{\text{ALARM}}$ (L), waarbij geldt L = 1 en H = 0.

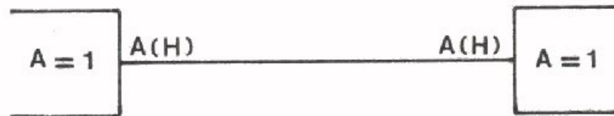
De volgende ALGEMENE REGEL is blijkbaar geldig:

In plaats van een signaal A (H) kan men ook een signaal $\overline{A}$ (L), en in plaats van signaal A (L) een signaal $\overline{A}$ (H) aanwezig veronderstellen

We zullen er ons in het vervolg aan houden om bij in- en uitgangen zonder polariteitsindicator steeds het (H)-signaal te noteren, en bij in- en uitgangen met polariteitsindicator het (L)-signaal.

DE NOT-FUNCTIE IN UITVOERINGSSCHEMA'S

Bij het met elkaar verbinden van twee schakelingen kunnen zich verschillende gevallen voordoen naargelang er al dan niet polariteitsindicatoren aanwezig zijn. We gaan deze gevallen stuk voor stuk bekijken.
Allereerst het geval zonder polariteitsindicator.



De linkerschakeling levert een signaal aan de rechterschakeling. Als in de linkerschakeling geldt A = 1, dan geeft de schakeling een hoge spanning af. Deze hoge spanning wordt door de rechter schakeling weer als signaal A = 1 gelezen. Er is niets bijzonders aan de hand.

Iets dergelijks treedt op als er aan beide zijden een polariteitsindicator aanwezig is.



Geldt links A = 1, dan levert de schakeling een lage spanning, die rechts weer als A = 1 wordt gelezen. Ook nu is er niets bijzonders aan de hand.

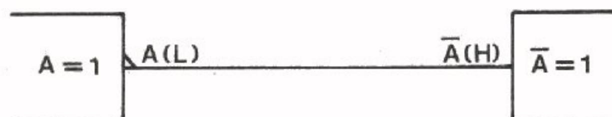
In het volgende geval is het echter anders.



De linkerschakeling geeft in geval van A = 1 een hoge spanning af. Deze hoge spanning wordt door de rechterschakeling echter als een 0 ervaren.
Er treedt hier blijkbaar een NOT-functie op, waarbij A wordt omgezet in $\overline{A}$.

De rechterschakeling levert A (H), waarbij geldt H = 1 en L = 0. In plaats van A (H) kunnen we ook $\overline{A}$ (L) noteren. Deze notatie moeten we voor de rechterschakeling gebruiken omdat deze een polariteitsindicator heeft, waar geldt L = 1 en H = 0.

Ook bij het laatste geval is er sprake van een NOT-functie.



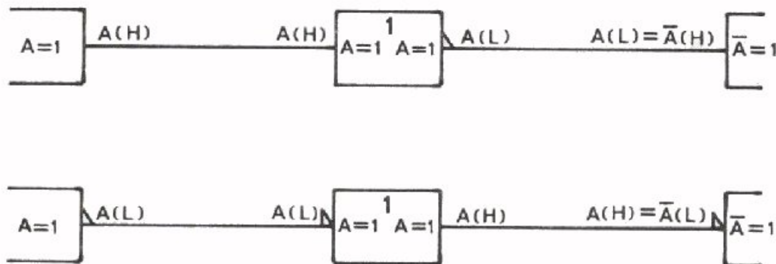
Beredeneer dit voor uzelf eens.

Uit deze 4 voorbeelden blijkt de volgende ALGEMENE REGEL:

> NEGATIE heeft altijd te maken met twee opeenvolgende schakelingen, waarbij de een de ander stuurt. Negatie treedt op als op hun onderlinge verbinding aan het ene einde een polariteitsindicator staat en aan het andere einde niet.

NEGATIE MET BEHULP VAN EEN OMKEERTRAP

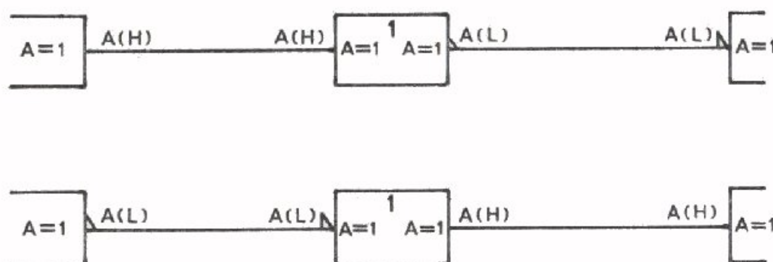
Zoals we hierboven hebben gezien, levert het verbinden van twee schakelingen die allebei wel of allebei niet van een polariteitsindicator voorzien zijn geen negatie op. Het kan echter zijn dat men in zo'n geval wel een negatie wenst. Die is dan te bereiken door het tussenschakelen van een omkeertrap. Dit blijkt uit de volgende figuren:



In deze gevallen bewerkt de omkeertrap dus een negatie. Dit is echter niet altijd het geval, zoals we hierna zullen zien. Daarom mogen we een inverter of omkeertrap dan ook NIET een negator of NOT-schakeling noemen.

GEEN INVERSIE M.B.V. OMKEERTRAP

Een verbinding tussen twee schakelingen met slechts een polariteits-indicator aan één kant betekent een negatie. Is een negatie op die plaats niet gewenst, dan kan die vermeden worden door het tussenschakelen van een omkeertrap. Dit blijkt uit de volgende figuren:

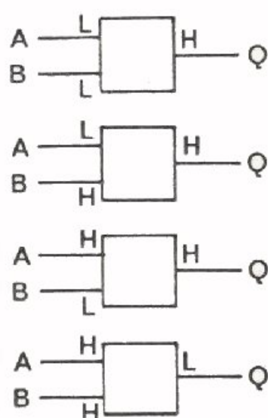


Uit de laatste gedeelten volgt duidelijk:

Omkeren behoeft nog geen negatie in te houden

DE INVERTERENDE AND-POORT (Z.G. NAND)

Een volgend voorbeeld van een digitale bouwsteen is een schakeling die om historische redenen vaak NAND-poort genoemd wordt. Vooral in de TTL-techniek is het een veelgebruikte schakeling. In de figuren (a) en (b) hebben we weer het fysische gedrag en de functietabel weergegeven.



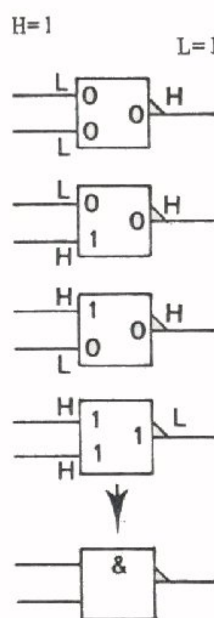
fysisch gedrag

(a)

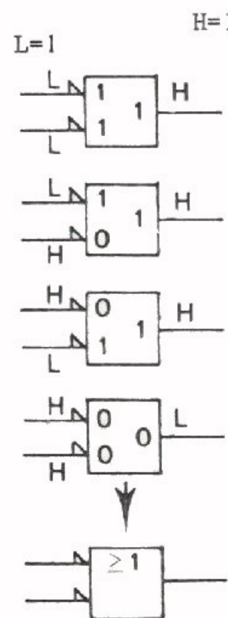
A	B	Q
L	L	H
L	H	H
H	L	H
H	H	L

functie-tabel

(b)

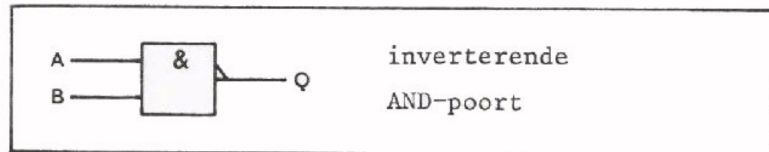


(c)



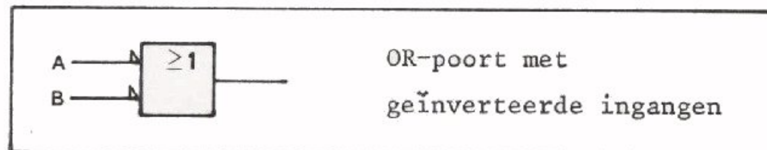
(d)

Uit de functietabel volgt, dat de uitgang alleen L is als alle ingangen H zijn. De functie kan daarom worden weergegeven door een AND-poort met een polariteitsindicator van de uitgang.



inverterende AND-poort

We lezen uit de tabel tevens af, dat de uitgang H is als een of meer ingangen L zijn. De functie kan dus ook worden weergegeven door een OR-poort met polariteitsindicatoren aan de ingangen.



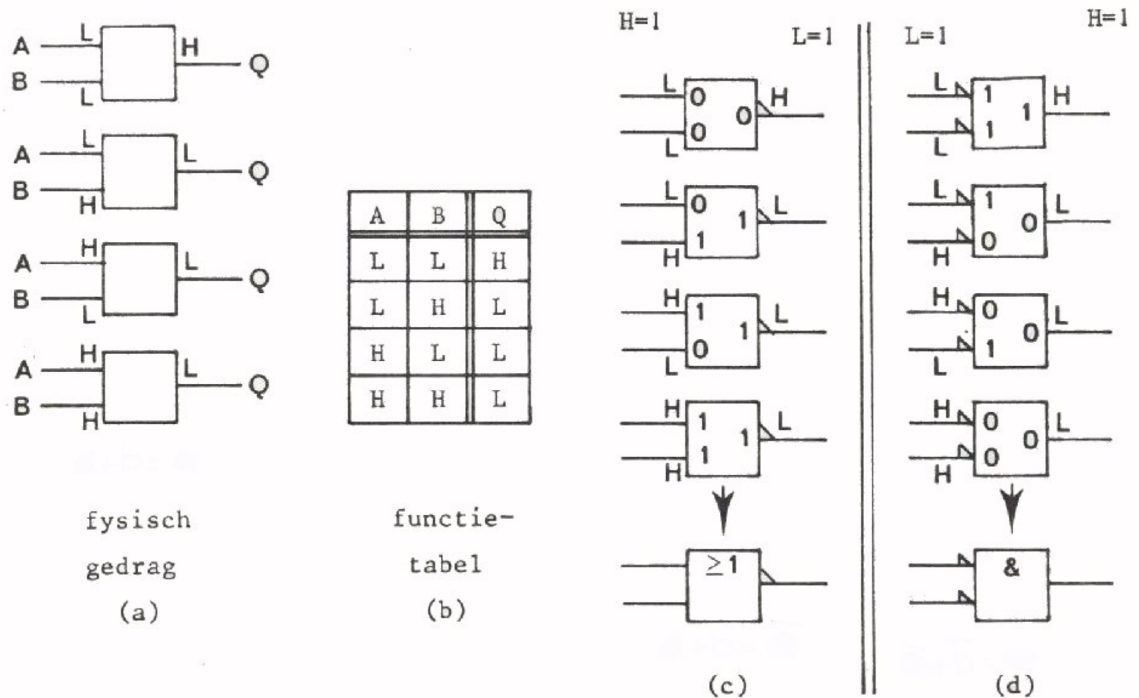
OR-poort met geïnverteerde ingangen

De bouwsteen is blijkbaar óf als inverterende AND-poort, óf als OR-poort met geïnverteerde ingangen op te vatten.

De algemene omzetregel voor combinatorische schakelingen blijkt weer op te gaan.

DE INVERTERENDE OR-POORT (Z.G. NOR)

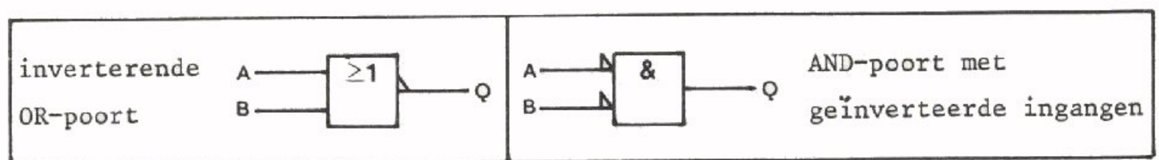
Het laatste bouwsteen voorbeeld is een inverterende OR-poort die vaak een NOR-poort wordt genoemd. Eerst weer het fysische gedrag en de functietabel in de figuren (a) en (b).



A	B	Q
L	L	H
L	H	L
H	L	L
H	H	L

fysisch gedrag (a) functie- tabel (b) (c) (d)

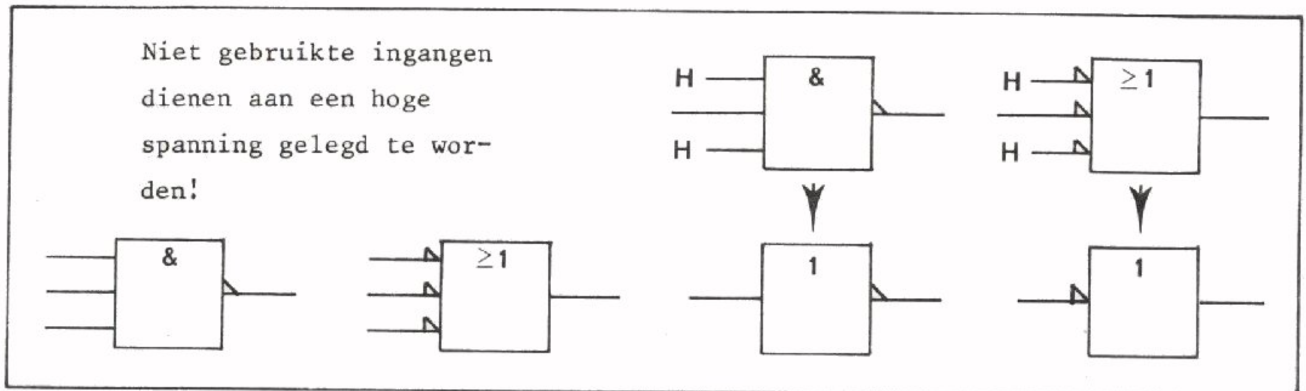
Op gelijke wijze als hiervoor kan men voor deze poort de volgende twee symbolen afleiden.



Ga dit voor uzelf na.

DE INVERTERENDE AND-REALISATIE (Z.G. NAND-REALISATIE)

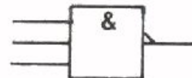
We gaan het ontwerpschema van de rolfilmautomaat volgens blz. 1.6 omzetten in een uitvoeringsschema, waarin inverterende AND's worden toegepast. De inverterende AND kunnen we ook als OR met geïnverteerde ingangen of als inverter gebruiken. Hij is over het algemeen op 4 manieren toe te passen. Deze zijn hieronder weergegeven.

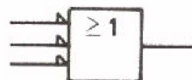


Met deze éne bouwsteen kunnen we dus de AND-functie, de OR-functie en de omkeertrap realiseren.

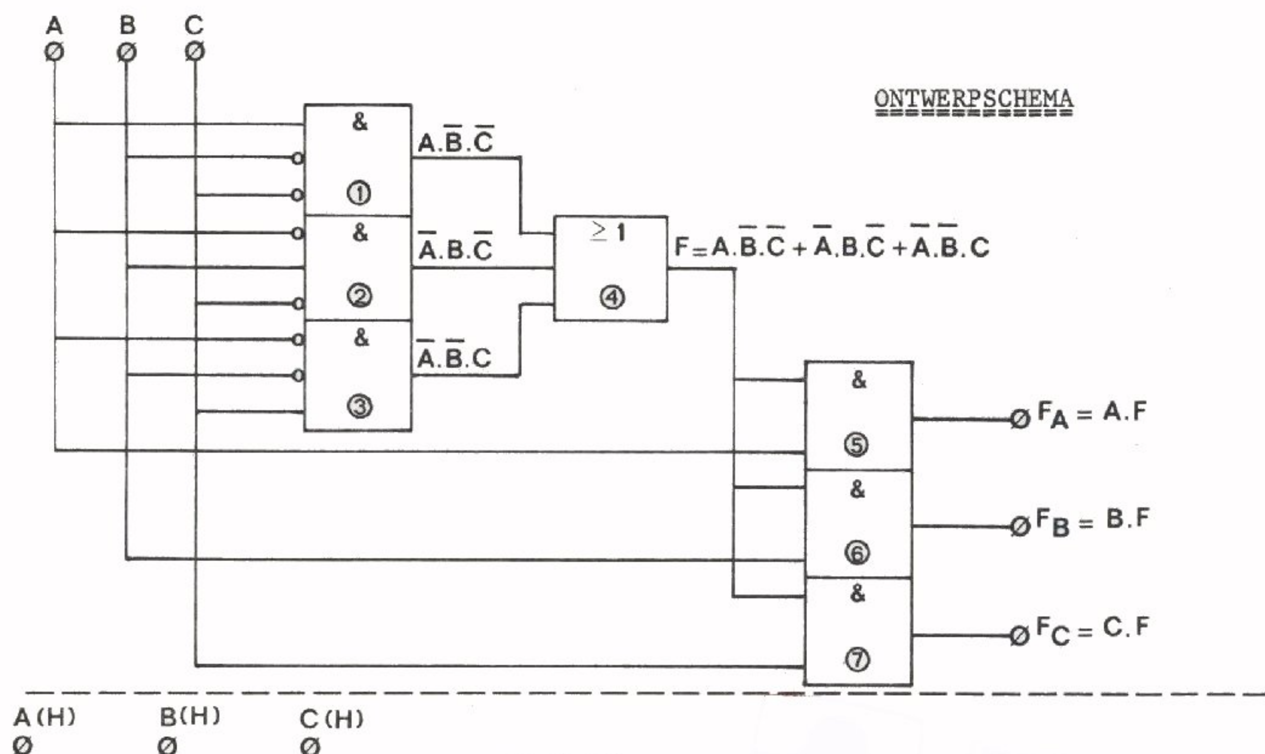
De omzetting van het ontwerpschema in het uitvoeringsschema kan men het beste als volgt verrichten (raadpleeg hierbij blz. 1.21):

- Teken eerst het ontwerpschema.
- Vervang daarna (voorlopig zonder doorverbindingen aan te brengen):

elke AND-poort door 

elke OR-poort door 

- Breng tenslotte de doorverbindingen aan. Levert die verbinding een negatie op (slechts aan één kant een polariteitsindicator), terwijl de negatie niet gewenst is, breng dan een omkeertrap aan. Levert de verbinding geen negatie op (aan beide kanten geen polariteitsindicatoren of aan beide kanten wel polariteitsindicatoren), terwijl er wel een negatie nodig is, breng dan een extra omkeertrap aan.



A (H)
B (H)
C (H)

POORTKEUZE

&
≥ 1

UITVOERINGSSCHEMA

A (H)
B (H)
C (H)
A (L)
C (L)
$\overline{A}$ (H)
$\overline{C}$ (H)
$A.\overline{B}.\overline{C}$ (L)
$\overline{A}.B.\overline{C}$ (L)
$\overline{A}.\overline{B}.C$ (L)
$F(H) = A.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + \overline{A}.\overline{B}.C$ (H)
H
A.F(L)
A.F(H)
B.F(L)
B.F(H)
C.F(L)
C.F(H)

Bij het uitvoeren van de 1e stap (het tekenen van de uitvoeringsschema-symbolen zonder doorverbindingen aan te brengen), dient men de symbolen niet te dicht op elkaar te tekenen. Dit om de doorverbindingen en eventueel in te lassen extra inverters naderhand gemakkelijk te kunnen aanbrengen.

Door de poorten van het ontwerpschema dusdanig te vervangen, dat de oorspronkelijke "& symbolen" en " $\geq$ 1 symbolen" behouden blijven, wordt het uitvoeringsschema erg begrijpelijk. Men blijft de oorspronkelijk bedoelde AND- en OR-functies namelijk direct zien.
Daarna komt de 2e stap, het tekenen van het volledige uitvoeringsschema.

- Volgens het ontwerpschema staan de ingangssignalen A, B en C ter beschikking. Aan de AND-functies (1), (2) en (3) moeten af en toe echter $\bar{A}$, $\bar{B}$ of $\bar{C}$ worden toegevoerd.
  Gaan we er vanuit dat A, B en C bij de praktische uitvoering als A(H), B(H) en C(H) ter beschikking staan, dan kunnen de tevens benodigde $\bar{A}$(H), $\bar{B}$(H) en $\bar{C}$(H) met behulp van drie inverters worden verkregen.
  De doorverbindingen van deze inverters met de ingangen van (1), (2) en (3) hebben slechts aan één zijde een polariteitsindicator. Daaraan kan men zien dat hier een negatie optreedt die ook in het ontwerpschema is aangegeven.

- De uitgangen van de poorten (1), (2) en (3) kan men direct met de ingangen van (4) doorverbinden. Dan ontstaan nl. verbindingen die aan beide einden polariteitsindicatoren hebben. Dit houdt in dat er geen negatie plaats vindt, zoals het ontwerpschema ook aangeeft.

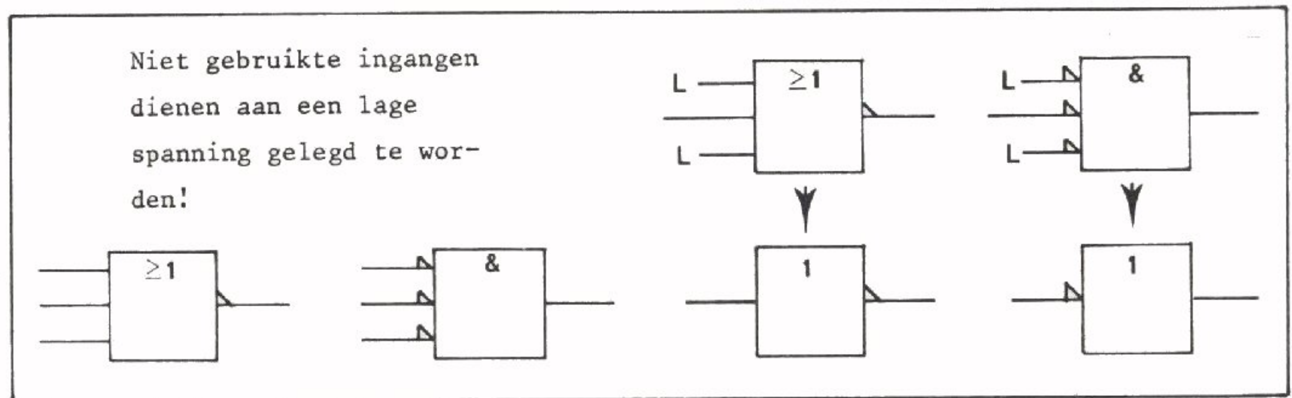
- De uitgang van poort (4) kan zonder meer worden doorverbonden met de ingangen van de poorten (5), (6) en (7). Ga voor Uzelf na waarom dit zo is.

- Bij het tekenen van omkeertrappen houden we de gewoonte aan het symbool steeds zó te tekenen, dat de negatie optreedt aan de kant van het symbool waar de negatie ook op het ontwerpschema is getekend. In ons voorbeeld is dit bij de ingangen van de poorten (1), (2) en (3). De omkeertrappen zijn nu zo getekend dat de negatie aan die kant optreedt. De ingangen van de poorten hebben geen polariteitsindicator; om negatie te krijgen moeten de uitgangen van de omkeertrappen dan wel een polariteitsindicator hebben.

- Willen we dat de uitgangssignalen H zijn als de betreffende voorwaarde vervuld is, dan moeten we achter elke uitgang nog een omkeertrap aanbrengen. De uitgangspoorten hebben polariteitsindicatoren aan de uitgangen; omdat hier geen negatie gewenst is, tekenen we dus de omkeertrappen met de polariteitsindicator aan de ingang.

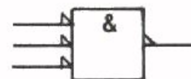
DE INVERTERENDE OR-REALISATIE (Z.G. NOR-REALISATIE)

Het rolfilmautomaten-ontwerp kunnen we op overeenkomstige wijze met behulp van inverterende OR-poorten realiseren.
De inverterende OR is op 4 manieren toe te passen:

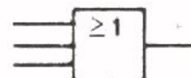


Op het volgende blad hebben we omzetting van het ontwerpschema weer in 2 stappen getekend. Bij de 1^{e} stap is:

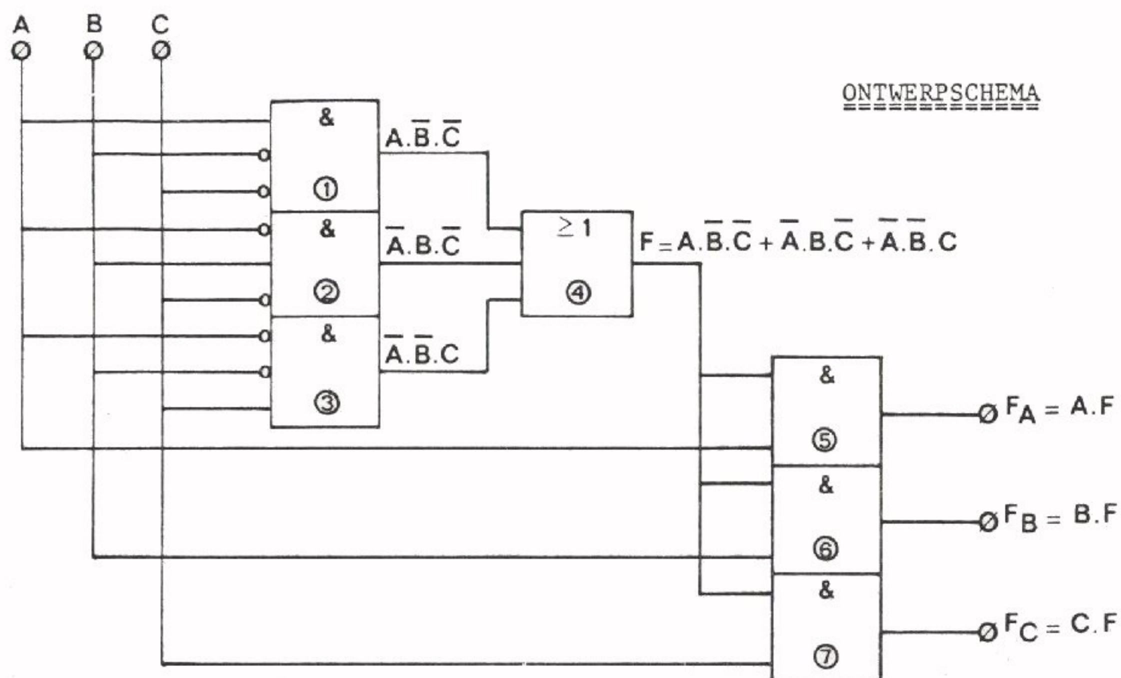
elke AND-poort door

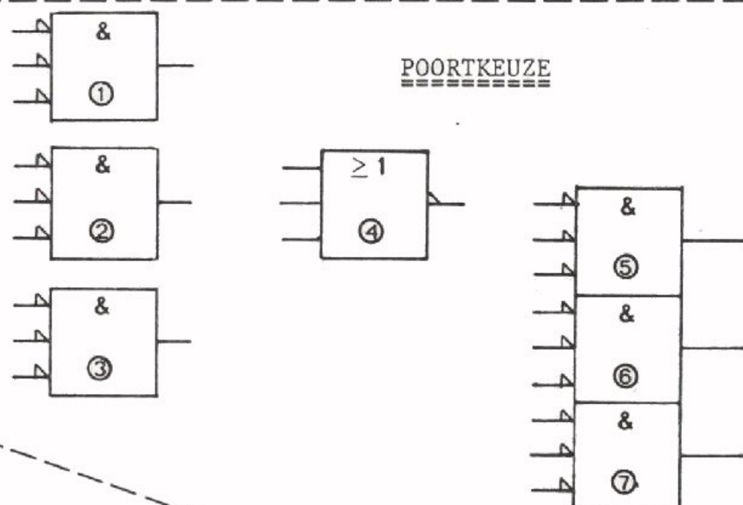


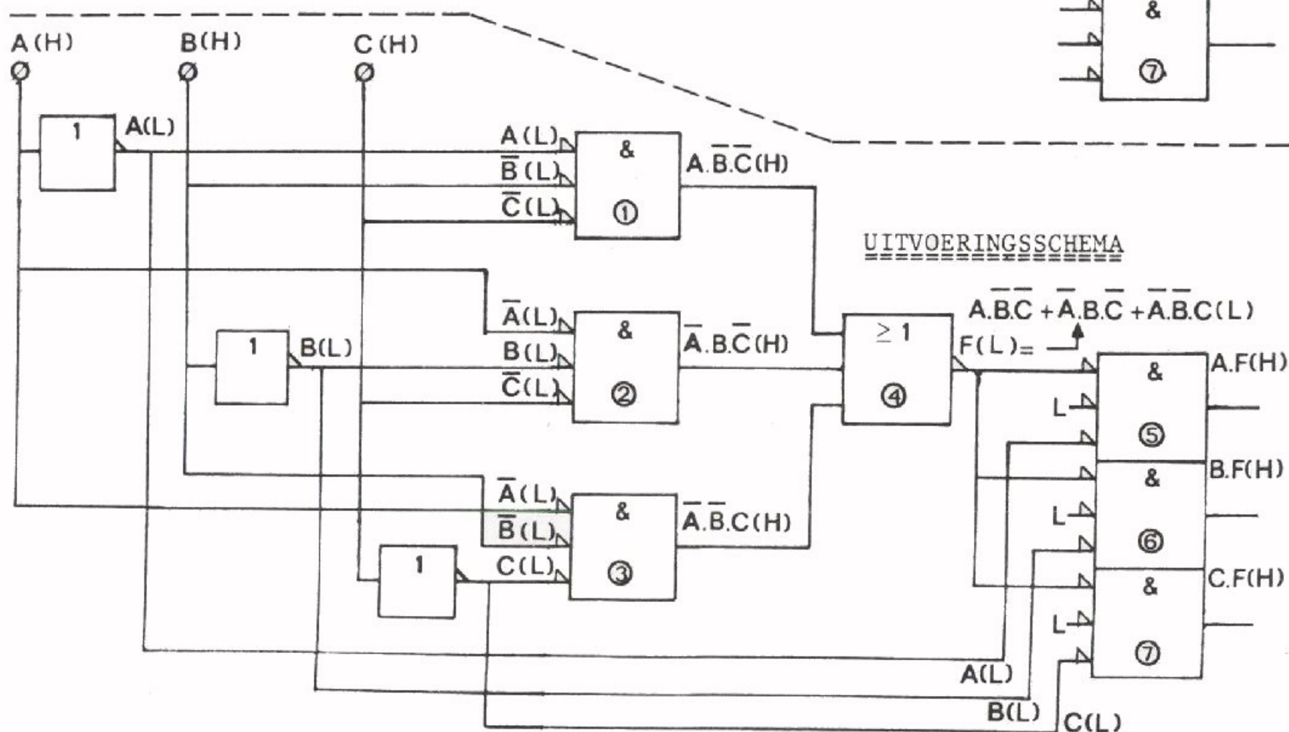
elke OR-poort door



vervangen.


ONTWERPSCHEMA
A
B
C
&
①
A.B̄.C̄
&
②
Ā.B.C̄
&
③
Ā.B̄.C
≥ 1
④
F = A.B̄.C̄ + Ā.B.C̄ + Ā.B̄.C
&
⑤
F_A = A.F
&
⑥
F_B = B.F
&
⑦
F_C = C.F


POORTKEUZE
&
①
&
②
&
③
≥ 1
④
&
⑤
&
⑥
&
⑦


UITVOERINGSSCHEMA
A(H)
B(H)
C(H)
1
A(L)
A(L)
B̄(L)
C̄(L)
&
①
A.B̄.C̄(H)
Ā(L)
B(L)
C̄(L)
&
②
Ā.B.C̄(H)
1
B(L)
Ā(L)
B̄(L)
C(L)
&
③
Ā.B̄.C(H)
1
≥ 1
④
A.B̄C̄ + Ā.B.C̄ + Ā.B̄.C(L)
F(L) =
&
⑤
A.F(H)
&
⑥
B.F(H)
&
⑦
C.F(H)
L
L
L
A(L)
B(L)
C(L)

Bekijk het uitvoeringsschema eens goed, daarbij voor Uzelf nagaand of elk detail duidelijk is! We maken er nog enkele opmerkingen over:

- Evenals in het vorige voorbeeld is ook hier aangenomen dat de ingangssignalen H zijn als de betreffende voorwaarden waar zijn., d.w.z. als de desbetreffende knoppen ingedrukt zijn.

- De omkeertrappen aan het begin hebben hier een andere functie dan in het vorige voorbeeld. Daar deden ze dienst om de gevraagde negatie te geven, hier dienen ze juist om de negatie te vermijden: zij hebben polariteitsindicatoren aan hun uitgangen en deze uitgangen sturen ingangen die eveneens van polariteitsindicatoren zijn voorzien.

- Het uitvoeringsschema is weer uiterst overzichtelijk: elke schakeling is zo getekend, dat er direct aan te zien is of hij als AND-functie, als OR-functie of als omkeertrap werkzaam is.

DE UITDRUKKINGEN "HOOG ACTIEF" EN "LAAG ACTIEF"

Een praktische bouwsteen geeft men weer met een uitvoeringsschemasymbool. Is een ingang (resp. uitgang) daarvan NIET voorzien van een polariteits-indicator, dan is de inwendige toestand 1 als uitwendig een hoge spanning (H) aanwezig is. Daarom noemt men die ingang (resp. uitgang) dan HOOG ACTIEF.
Is een ingang (resp. uitgang) WEL voorzien van een polariteits-indicator, dan is de inwendige toestand 1 als uitwendig een lage spanning (L) aanwezig is. Daarom noemt men die ingang (resp. uitgang) dan LAAG ACTIEF.

SAMENVATTING

Met behulp van de 3 basisfuncties AND, OR en NOT kan men elke digitale schakeling opbouwen.

Het SYSTEEM VAN DE ONTWERPSCHEMA-SYMBOLEN is bedoeld om gebruikt te worden bij het ontwerpen van digitale schakelingen, indien de latere praktische uitvoering nog niet bekend is.
Bij dit systeem houdt men zich aan volgende afspraken:

- De NOT-functie wordt altijd buiten het kader van het schemasymbool gehouden en wordt vertegenwoordigd door een negatie-indicator.
- De overige functies worden altijd binnen het kader gehouden.
- Zowel binnen als buiten de kaders worden de signaalwaarden aangeduid met 1 en 0 ("waar" en "niet waar").
- Een negatie-indicator zet 0 om in 1, of omgekeerd 1 om in 0.

Het verdient aanbeveling om negaties niet als aparte NOT-elementen te tekenen, maar ze als negatie-indicatoren aan ingangen of uitgangen van aangrenzende digitale elementen te zetten.

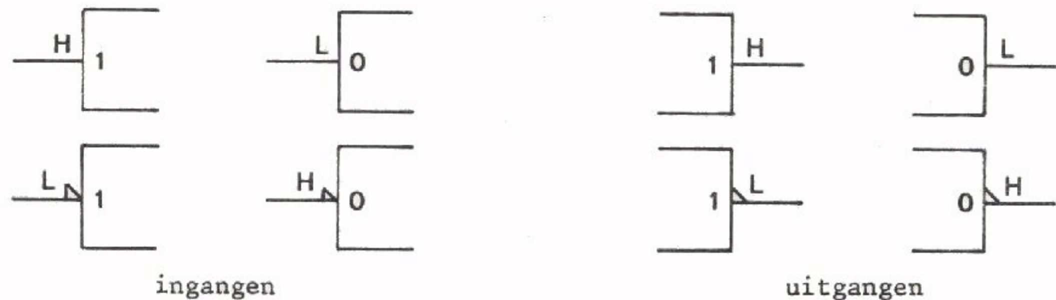
Het gedrag van een digitale schakeling is aan de hand van het ontwerpschema vast te leggen in een waarheidstabel en/of volgordetabel. In de tabellen worden 0 en 1 ingevuld.

- Elke regel van een WAARHEIDSTABEL is op zichzelf altijd waar, onverschillig wat er aan de vermelde toestand is voorafgegaan.
- Een regel van een VOLGORDETABEL behoeft niet op zichzelf altijd waar te zijn omdat de voorafgaande toestand soms van invloed is. Zodra een digitale schakeling een of meer geheugenelementen bevat, heeft de voorgaande toestand soms invloed op de volgende.

Bij het SYSTEEM VAN UITVOERINGSSCHEMA-SYMBOLEN houdt men zich aan volgende afspraken:

- De negatie wordt altijd buiten het kader van het schemasymbool gehouden.
- De invertering wordt altijd buiten het kader van het schemasymbool gehouden.
- De overige functies worden altijd binnen het kader gehouden.
- Binnen de kaders worden de signaalwaarden aangeduid met 1 en 0 ("waar" en "niet waar").
- Buiten de kaders worden de signaalwaarden aangeduid met H en L ("Hoge spanning" en "Lage spanning").

- Tekent men géén polariteitsindicator, dan gaat H buiten het kader samen met 1 binnen het kader en L buiten met 0 binnen.
  Tekent met wel een polariteitsindicator, dan gaat L buiten het kader samen met 1 binnen het kader en H buiten met 0 binnen.

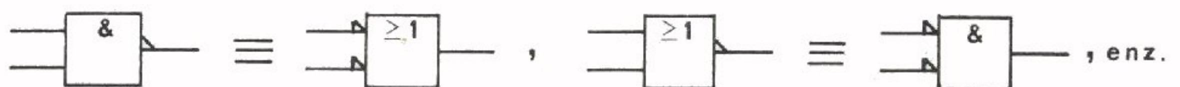


- In een uitvoeringsschema kan dán weer gelden H = 1 en L = 0, dán weer L = 1 en H = 0.
  Geldt H = 1 en L = 0, dan zet men (H) achter het signaalsymbool, resp. de signaalformule.
  Geldt L = 1 en H = 0, dan zet men (L) achter het signaalsymbool, resp. de signaalformule.

Is op een punt signaal A(H) aanwezig, dan kan men in plaats daarvan ook $\overline{A}$(L) aanwezig veronderstellen. Net zo is in plaats van A(L) het signaal $\overline{A}$(H) te stellen.

ALGEMENE OMZETREGEL: Iedere combinatorische digitale bouwsteen kan op twee manieren getekend worden. Men mag & vervangen door ≤1 en omgekeerd als men "de in- en uitgangen zonder polariteitsindicator" van een polariteitsindicator voorziet en omgekeerd van "de in- en uitgangen met polariteitsindicator" de polariteitsindicatoren weglaat.

Dit houdt in dat elke bouwsteen op twee manieren is te zien:



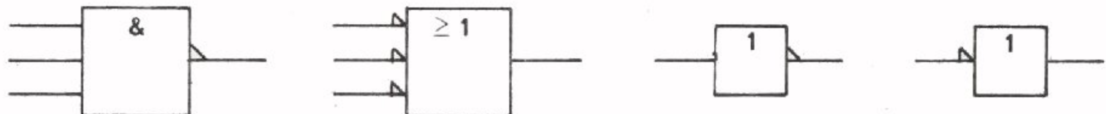
Bevindt zich aan het ene eind van een verbinding een polariteitsindicator en aan het andere eind niet, dan treedt negatie van het signaal op.
Bevinden zich aan beide einden van een verbinding polariteitsindicatoren, of aan beide einden niet, dan treedt er geen negatie op.

De FUNCTIETABELLEN en de VOLGORDETABELLEN van uitvoeringsschema-symbolen komen overeen met de "waarheidstabellen" en de "volgordetabellen" van ontwerpschema-symbolen.
In uitvoeringsschematabellen staan niet de waarden 1 en 0, maar de waarden H en L. Met deze tabellen geeft men het FYSISCHE GEDRAG van de bouwstenen weer: wat voor spanning er uitkomt als men bepaalde spanningen toevoert.

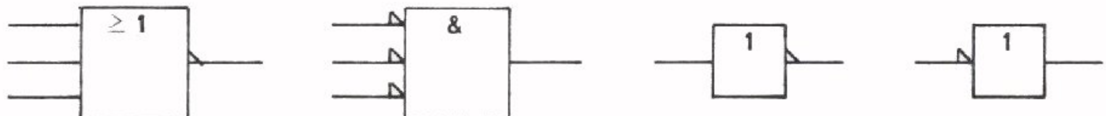
Inverteren of omkeren is het omzetten van een H-niveau in een L-niveau en omgekeerd.

De namen AND, OR, NAND en NOR zijn eigenlijk onjuist, want elke poort kan zowel een AND- als een OR-functie vervullen.

Een inverterende AND-poort (z.g. NAND) is te zien als:



Een inverterende OR-poort (z.g. NOR) is te zien als:



<u>Het uitvoeringsschema is direct uit het ontwerschema af te leiden</u> door:

1e. De poorten van het ontwerpschema in uitvoeringsvorm over te tekenen, daarbij telkens het poortsymbool zó kiezend dat de functie volgens het ontwerpschema naar voren komt (zie voorgaande punten omtrent inverterende AND en OR).

2e. Daarna de verbindingen te tekenen, zonodig extra INVERTERS inpassend, zodat er negaties ontstaan waar ze gewenst zijn en negaties vermeden worden waar ze niet gewenst zijn.

De gunstige eigenschappen van het behandelde uitvoeringsschema-systeem zijn:

- Op elk punt van het uitvoeringsschema is direct te zien of een hoge dan wel een lage spanning aanwezig moet zijn opdat het signaal "waar" (= 1) is.

- Men kan de signalen goed door de schakeling heen volgen omdat telkens direct te zien is of er een AND-functie of een OR-functie bedoeld is en of er van een negatie sprake is.

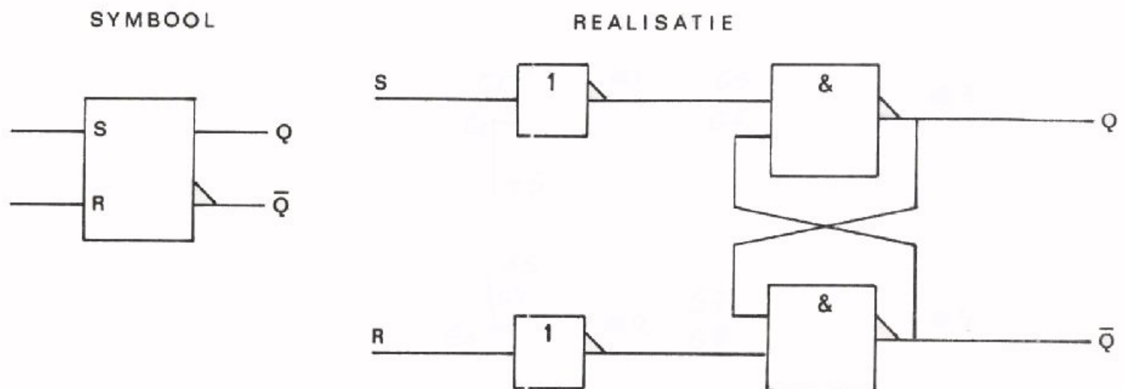
2 F L I P - F L O P - R E G I S T E R S

INLEIDING

In digitale systemen nemen schakelingen die "onthouden" in welke toestand zij geplaatst werden een zeer belangrijke plaats in. In deel D van de cursus Bedrijfselektronica hebben we verschillende van deze schakelingen uitvoerig besproken. Deze zijn o.a. de SR-flip-flop, de geklokte SR-flip-flop en de master-slave JK-flip-flop. Al deze schakelingen kunnen met behulp van inverterende AND of OR poorten samengesteld worden. In deze les gaan we de SR-flip-flop en de geklokte SR-flip-flop nog eens bekijken met de bedoeling twee nieuwe flip-flops af te leiden: de D-flip-flop en de latch-flip-flop. Daarna zullen we zien hoe men registers kan opbouwen met behulp van deze flip-flops.

DE SR-FLIP-FLOP

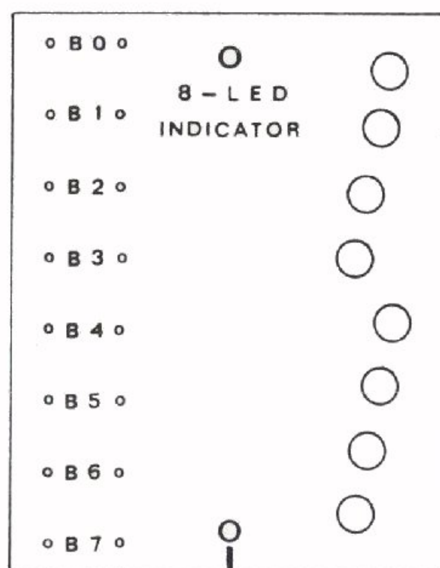
Hieronder vindt U links het symbool van een SR-flip-flop. Rechts vindt U de realisatie ervan met behulp van een 7400 module.



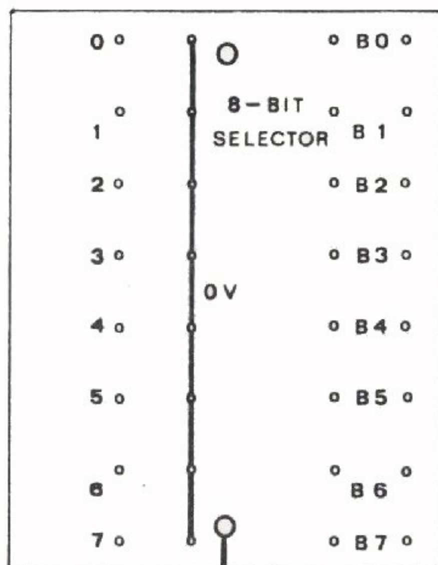
Aan het gebruik van de letters S en R in het symbool is te zien dat het een SR-flip-flop is. De realisatie bevat aan elk van de ingangen een inverterende poort. Hiervoor worden de twee resterende poorten van de 7400 module gebruikt.

DE 8-LED INDICATOR EN DE 8-BIT SELECTOR

Alvorens een proef uit te voeren met de zo juist beschreven SR-flip-flop, gaan we twee nieuwe modulen introduceren. Deze modulen, de 8-LED indicator en de 8-BIT selector, zult U in het vervolg van deze cursus regelmatig gebruiken.



Hiernaast ziet U een afbeelding van de bovenplaat van de 8-LED indicator. B0 t/m B7 zijn de ingangen. Wanneer een van deze ingangen is aangesloten aan de 0 V (= L), brandt de overeenkomstige LED niet. Wanneer een ingang aangesloten is aan de 5 V (= H) brandt de LED wel.



De 8-BIT selector kan beschouwd worden als een schakeling, waarmee men een willekeurige combinatie van 8-bits kan maken door al of niet doorverbindingen aan te brengen.

Wanneer ingang 0 met behulp van een kortsluitbeugel aan de 0 V verbonden is, wordt de uitgang B0 hoog (H). Wordt de kortsluitbeugel verwijderd dan is de uitgang B_0 laag (L). Wordt ingang 1, 2 ... verbonden met behulp van een kortsluitbeugel met de 0 V, dan wordt de betreffende uitgang B_1, B_2 ... hoog (H). Bij verwijdering van de beugel is de uitgang dan weer laag (L).

Samenvattend: aangenomen dat de logische waarde "1" wordt gerepresenteerd door hoog (H), dan kunnen we zeggen dat de aanwezigheid van een kortsluitbeugel aan een ingang overeenkomt met een "1" op de desbetreffende uitgang. De voedingsspanning van beide modulen bedraagt 5 V. Deze wordt toegevoerd via de onderste contactpennen.

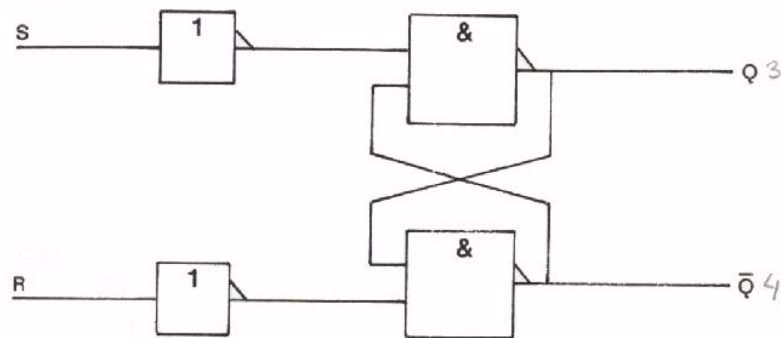
OPDRACHT 2.1: METINGEN AAN DE 8-BIT SELECTOR EN DE 8-LED INDICATOR MODULEN

- Prik op het schakelpaneel een 5 V voedingsstabilisator, een 8-BIT selector en een 8-LED indicator.

- Verbind de B uitgangen van de 8-BIT selector met de overeenkomstige ingangen van de 8-LED indicator.

- Wanneer U een ingang van de 8-BIT selector doorverbindt met de 0 V met behulp van een kortsluitbeugel, gaat de overeenkomstige LED van de indicator aan.

OPDRACHT 2.2



Bouw bovenstaande SR-flip-flop met behulp van een 7400-module op het schakelpaneel.

Verbind de ongebruikte ingangen van de inverterende AND's met + 5 V.

Meet de toestanden van Q en $\bar{Q}$ voor de combinaties van S en R die zijn aangegeven in de volgordetabel.

Breek de schakeling nog niet af.

S	R	Q	$\bar{Q}$
L	L		
L	H		
H	L		
L	L		
L	H		
H	H		
H	L		
L	L		

CONCLUSIES

Kenmerkend voor deze SR-flip-flop is:

- Stuursignaal "H". Door S = H <u>of</u> R = H te maken wordt de flip-flop geSet of geReset.
- Onthoudtoestand: S = L <u>en</u> R = L.
  Door S = L en R = L te maken wordt de vooraf ingebrachte informatie onthouden.
- Vergeettoestand: S = H <u>en</u> R = H.
  Door S = H en R = H te maken wordt de vooraf ingebrachte informatie vergeten.

Bekijk dit nog eens aan de hand van de volgordetabel.

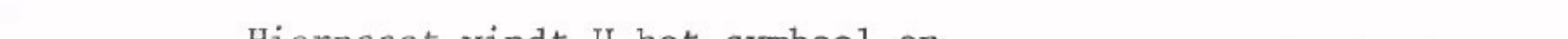
OPMERKING

In de SR-flip-flop worden de combinaties

S = L en R = H

of S = H en R = L

direct door de uitgangen overgenomen. Er zijn SR-flip-flops waaraan een kloksignaal-ingang is toegevoegd. We beperken ons daarbij tot de "geklokte" SR-flip-flop met 4 inverterende AND poorten.



OPDRACHT 2.3: DE GEKLOKTE SR-FLIP-FLOP

Hiernaast vindt U het symbool en de realisatie van een uit vier inverterende AND's samengestelde geklokte SR-flip-flop.
Wijzig de schakeling van de SR-flip-flop zoals hieronder weergegeven.



Voer de signalen toe, zoals ze zijn weergegeven in de volgordetabel en noteer welke invloed zij hebben op uitgang Q en $\overline{Q}$.

Breek de schakeling nog niet af.

a	b	T	Q	$\overline{Q}$
H	L	L		
H	L	H		
H	L	L		
L	L	L		
L	H	L		
L	H	H		
L	H	L		
L	L	L		
L	H	H		
L	L	H		
H	L	H		
L	L	H		
L	L	L		
H	H	H		
H	H	L		

CONCLUSIES

Uit de volgordetabel blijkt:

Q wordt H als a = H en b = L en T = H

Q wordt L als a = L en b = H en T = H

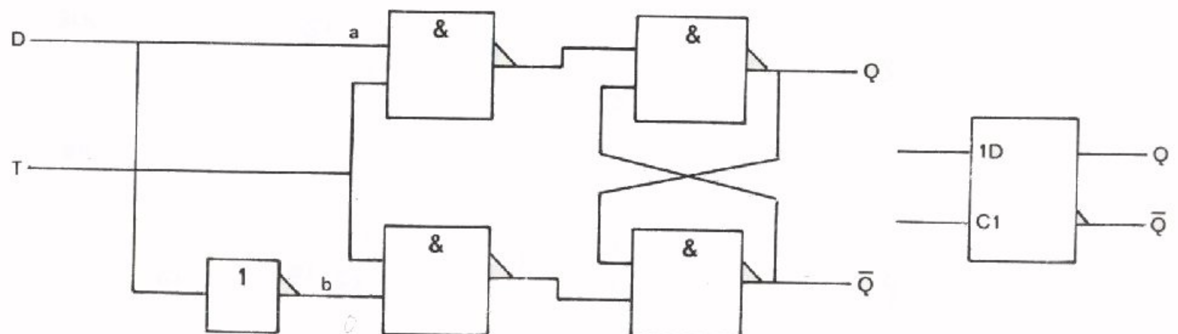
Bovendien zien we dat het uitgangssignaal Q verandert met het ingangssignaal zolang T hoog gehouden wordt en verandert niet meer wanneer ingang T laag blijft.

Met andere woorden:

- De uitgang Q van een geklokte SR-flip-flop wordt gelijk aan ingang a als ingang b de inverse is van a en het kloksignaal T is H.
- Veranderen a en b terwijl T = L, dan blijft de informatie op Q gehandhaafd op de voorgaande informatie. Bij T = L, is de "onthoudtoestand" aanwezig waarin de set- of reset-toestand wordt onthouden.
- Door a = H, b = H en T = H toe te voeren ontstaat de vergeettoestand Q = H en $\overline{Q}$ = H. Wordt daarna T = L, dan is vergeten wat voor de laatste T = H aanwezig was.

DE LATCH-FLIP-FLOP

Zoals we reeds hebben gezien, heeft de geklokte SR-flip-flop, naast de klokingang T, twee ingangen a en b nodig die in de combinatie LH of HL aanwezig moeten zijn. Bovendien kan er een vergeettoestand optreden. Wanneer de ingangsinformatie allēēn in de vorm L of H aanwezig is, moet een inverterende poort toegevoegd worden, zoals hieronder weergegeven.



Aan de voorwaarde b = $\overline{a}$, wordt nu altijd voldaan, zodat er gēēn vergeettoestand meer kan optreden.

OPDRACHT 2.4: HET METEN AAN EEN LATCH-FLIP-FLOP

Wijzig de schakeling van de vorige opdracht door toevoeging van een inverterende poort. Gebruik hiervoor een van de vier inverterende AND poorten van de tweede 7400 module. Denk er goed aan dat de ongebruikte ingang van de gekozen poort met 5 V verbonden moet zijn.

Voer de signalen aan de D-ingang en aan de klokingang toe zoals aangegeven in de volgordetabel hiernaast.
Vul de tabel in.

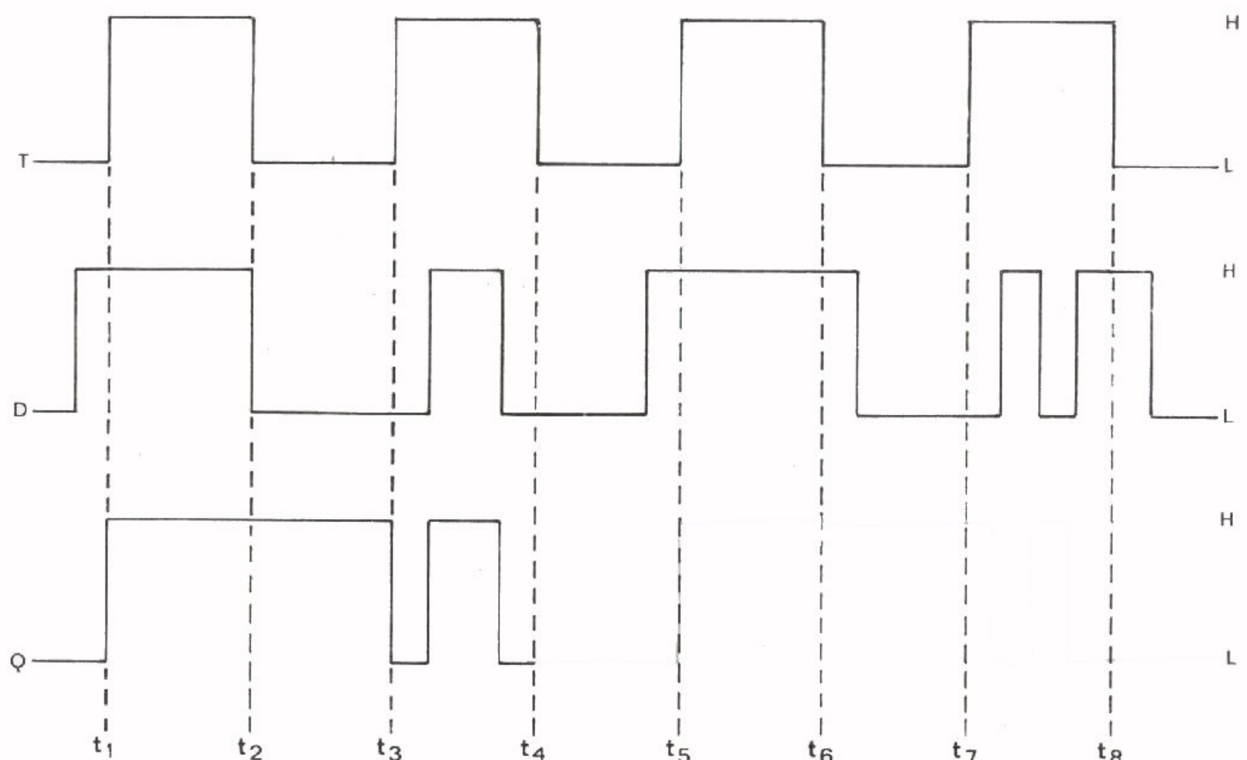
Breek nu de schakeling af.

D	T	Q	$\overline{Q}$
L	L		
H	L		
H	H		
H	L		
L	H		
H	H		
L	H		

CONCLUSIE

Uit de volgordetabel zien we dat de Q-ingang H wordt indien D = H en T = H. Als daarna T = L wordt, blijft Q = H. Deze laatste is dan de onthoudtoestand. De flip-flop die we nu gekregen hebben wordt een latch-flip-flop genoemd. Deze latch-flip-flop, zoals we nu hebben beschreven, is een schakeling die goed hanteerbaar is in digitale systemen omdat de logische niveaus van de uitgang slechts afhankelijk zijn van één signaal op de D-ingang, dat aanwezig moet zijn gedurende de tijd dat de klokimpuls op de T ingang hoog is. Bovendien is er geen vergeettoestand mogelijk.

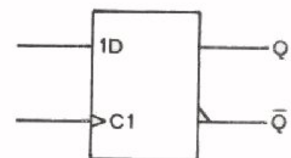
De niveaus kunnen we nog in een grafiek weergeven.

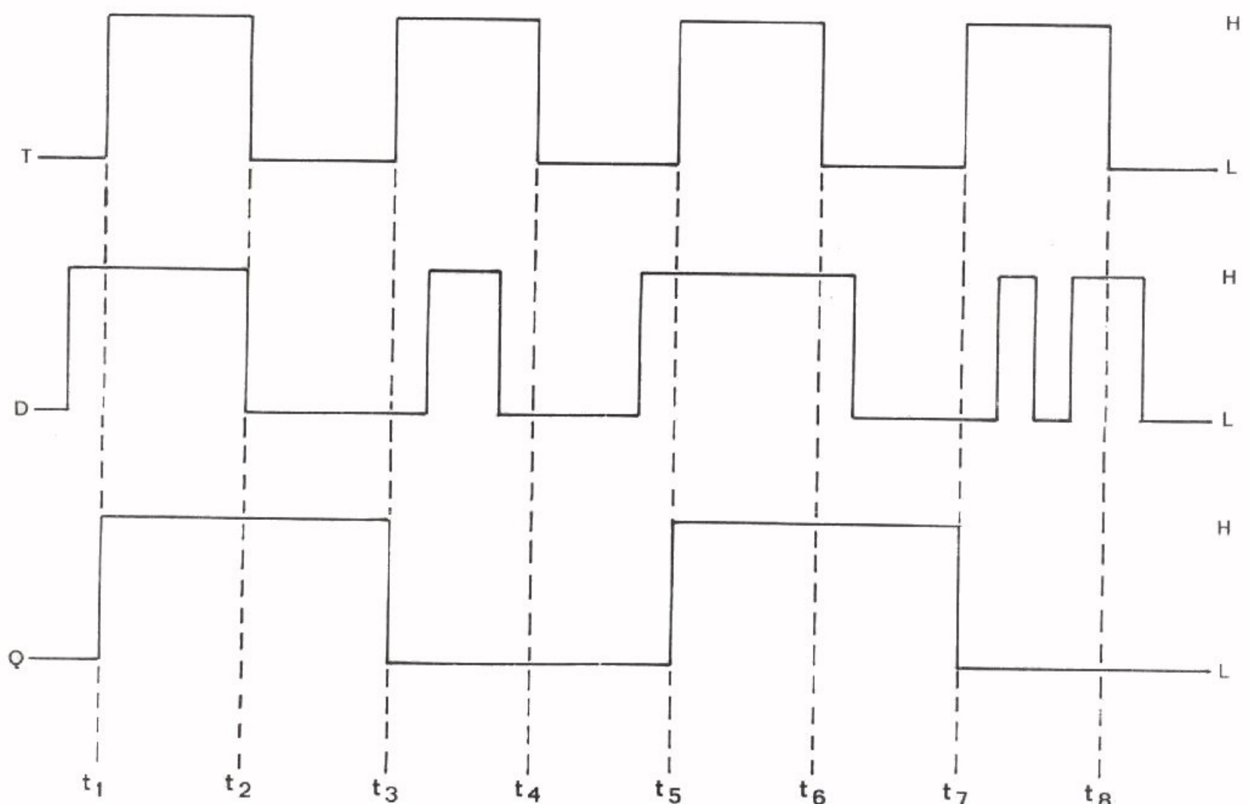


Op tijd t_1 worden de T- en de D-ingangen tegelijkertijd H. De uitgang Q wordt H en blijft H tot t_3. Tussen t_2 en t_3 blijft de latch-flip-flop in de onthoudtoestand want T = L.
Op tijd t_3, is D = L en T = H; de uitgang Q wordt dus L. Tussen t_3 en t_4 zal de uitgang Q de waarde van D volgen, want T is steeds H.
Ga zelf na welke toestand de uitgang Q zal aannemen tussen de tijden t_4 en t_8.

D-FLIP-FLOP

We hebben gezien dat bij een latch-flip-flop de uitgang Q de toestand van de D-ingang volgt zolang de klokimpuls aanwezig is. De D-flip-flop heeft, evenals de latch-flip-flop, slechts 1 ingang. De latch-flip-flop wordt daarom ook vaak met de D-flip-flop verward. De D-flip-flop heeft echter een dynamische klokingang, zodat de uitgangen veranderen als de klokimpuls van L naar H gaat, dus op de voorflank van de klokimpuls. De D-flip-flop wordt daarom een "edge triggered" (edge = flank) flip-flop genoemd. Hiernaast is het symbool van een D-flip-flop getekend. De werking van een D-flip-flop is hieronder in een grafiek weergegeven. Vergelijk deze grafiek met die van de latch-flip-flop en probeer zelf de verschillen te verklaren in de niveau's van de Q uitgang voor beide flip-flops.



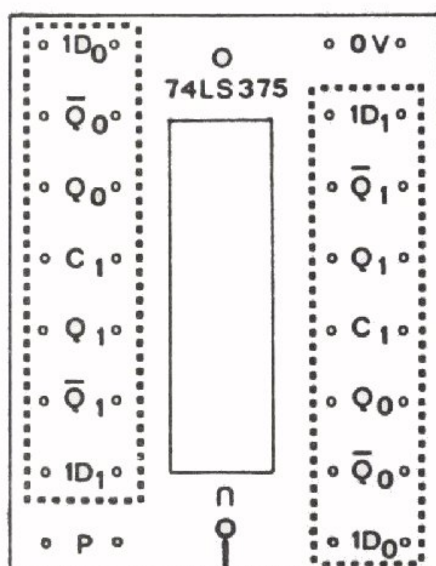


Zowel latch-flip-flops als D-flip-flops zijn veel gebruikte schakelingen in de digitale techniek. Ze bestaan als zodanig in IC vorm.

Bijvoorbeeld:

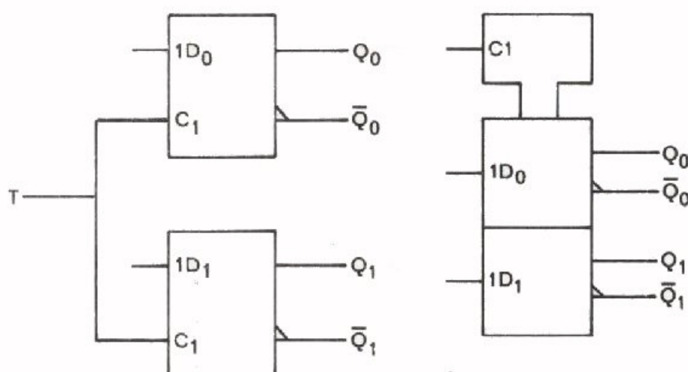
- de 74LS377 is een achtvoudige D-flip-flop. Hij bevat acht identieke flip-flop's die met dezelfde klokimpuls getriggerd worden. De informatie aan een D-ingang wordt doorgegeven aan de bijbehorende Q uitgang als de klokimpuls van L naar H gaat.
- de 74LS375 is een viervoudige latch-flip-flop. Dit IC bevat vier identieke flip-flops.

DE 74LS375 MODULE



Hiernaast zijn in bovenaanzicht de plaatsen van de contacten van de 74LS375-module aangegeven. De aansluitingen komen overeen met die op het IC-blokje.

Het bovenste 0 V -contact is doorverbonden met de onderste contactpennen op de bodem. Het P-contact linksonder is doorverbonden met de bovenste contactpennen. De voedingsspanning bedraagt 5 V.



De 74LS375 bevat vier latch-flip-flops. Ze hebben twee aan twee een gemeenschappelijke klokingang, zoals hiernaast in het linker circuit is weergegeven.

Dit circuit kan vereenvoudigd weergegeven worden met het rechter symbool. Dit symbool betekent dat er twee latch-flip-flops aanwezig zijn (twee onderste kaders).

Tussen de twee latch-flip-flops bestaat er geen logische verbinding (één horizontale lijn tussen de twee kaders) in de zin dat de ingang of uitgangen van de ene de in- of uitgangen van de andere niet beïnvloeden.

Het bovenste blok wordt het "Commandoblok" genoemd. Het bevat de gemeenschappelijke ingang van de flip-flops beschreven in het onderste blok.

OPDRACHT 2.5: METINGEN AAN DE LATCH-MODULE 74LS375

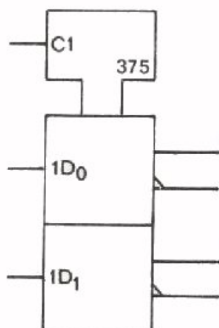
- Zet een latch-module op het schakelpaneel.
- Sluit de ingangen $1D_0$, $1D_1$ en de klokingang C_1 aan op de 8-BIT selector en de uitgangen Q_0 en Q_1 aan op de 8-LED indicator module.

D_0	D_1	C_1	Q_0	Q_1
L	L	H		
H	L	H		
L	H	H		
H	H	H		
L	H	L		
L	L	L		
H	L	H		
H	L	L		
H	H	H		
H	H	L		

Voer aan de ingangen signalen toe zoals aangegeven in nevenstaande tabel. Vul voor iedere mogelijkheid de toestand van Q_0 and Q_1 in. Herhaal daarna de opdracht voor de twee andere latch-flip-flops.

CONCLUSIES

Verbinden we van de latch-flip-flops de klokingangen met elkaar, dan zullen als deze ingangen hoog zijn de uitgangen Q de desbetreffende ingang D volgen. Zijn de klokingangen laag, dan onthoudt de schakeling de informatie totdat nieuwe informatie wordt ingeklokt.



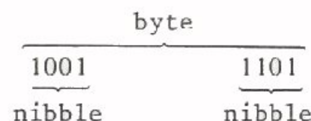
De helft van de 74LS375 module is hiernaast nog eens weergegeven. U heeft gezien dat deze schakeling twee binaire cijfers kan onthouden. In feite hebben we hier een 2-bits register gebouwd met behulp van latch-flip-flops. In deel D van de cursus hebben we registers bestudeerd die opgebouwd waren uit JK master-slave flip-flop's.

REGISTERS

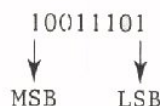
Een register is een uit flip-flops samengestelde digitale schakeling en het dient om informatie tijdelijk te bewaren. Het is een geheugen voor informatie. Wanneer een register wordt gebruikt om parallel informatie op te nemen en te onthouden totdat deze informatie weer in parallel wordt gevraagd, is het vaak opgebouwd uit D-flip-flop's of uit latch-flip-flops. Het IC 74LS375 bevat 4 latch-flip-flop's; het is dus mogelijk met behulp van dit IC een binair getal van 4 bits te onthouden.
De inhoud van een register wordt meestal "woord" genoemd, omdat de bij elkaar behorende bits niet altijd een getal behoeven voor te stellen, maar bijvoorbeeld ook een opdracht. Dit zal in een volgende les duidelijk worden.

We hebben over "bit" gesproken. Deze afkorting van het Engels "BInary digiT" werd reeds in deel D begruikt. Een bit is een "1" of een "0" die een bepaalde plaats of positie inneemt in een binair getal. Elke bit van een binair getal is het cijfersymbool "0" of "1".
Er zijn nog andere namen die in samenhang met registers worden gebruikt. We zullen ze nu reeds introduceren. Een "byte" is een rij van, bijvoorbeeld, acht bits. Een "nibble" is een rij van vier bits.

Het getal 10011101 is een byte en bevat twee nibbles



Vaak praten we ook over de "Least Significant Bit" afgekort LSB. Het is de bit met het kleinste gewicht. De Most Significant Bit, afgekort MSB, is de bit met het grootste gewicht. Voorbeeld:

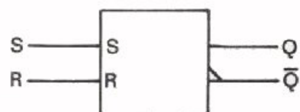


WAARHEIDSTABELLEN, VOLGORDETABELLEN EN FUNCTIETABELLEN

Tijdens de opdrachten die betrekking hadden op flip-flops hebben we gebruik gemaakt van volgordetabellen. Met deze tabellen kunnen we het gedrag van een element bepalen door aan zijn ingangen, in een bepaalde volgorde, het niveau hoog of laag aan te geven.

Voor de fundamentele functies (AND, OR en NOT), hebben we meestal gewerkt met waarheidstabellen die worden uitgedrukt in de logische waarden 0 en 1. Een waarheidstabel geeft alle mogelijke combinaties van de ingangsvariabelen met hun invloed op het resultaat, de uitgangsvariabelen. In deel D hebben we gezien dat een waarheidstabel 2^n regels bevat, waarin n het aantal ingangsvariabelen is. In een volgordetabel is het aantal regels bepaald door de handelingen die gewenst zijn om een effect duidelijk te maken. In dit laatste geval, zal het aantal regels meestal groter zijn dan in een waarheidstabel.

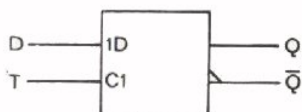
Om de functie van een flip-flop te kunnen beschrijven gebruiken we een functietabel. Hieronder vindt U het symbool en de functietabel van een SR-flip-flop.



S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
L	L	Q_n	$\overline{Q}_n$	1
L	H	L	H	2
H	L	H	L	3
H	H	H	H	4

- Regel 1 is de onthoudtoestand. De toestand Q_{n+1} van uitgang Q op het moment n + 1 is gelijk aan de vooraf, op moment n, ingebrachte informatie (Q_n).
- Regel 2 is de reset toestand.
- Regel 3 is de set toestand.
- Regel 4 is de vergeettoestand.

Hetzelfde kunnen we doen voor een latch-flip-flop. Hieronder geven we het symbool en de functietabel.

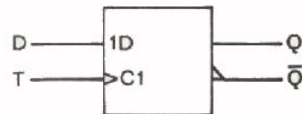


T	D	Q_{n+1}	$\overline{Q}_{n+1}$
L	L	Q_n	$\overline{Q}_n$
L	H	Q_n	$\overline{Q}_n$
H	L	L	H
H	H	H	L

● De toestand van de uitgangen blijft ongewijzigd als het kloksignaal T laag is.

● De uitgang Q volgt de ingang D als het kloksignaal hoog is.

Tot slot nog het symbool en de functietabel van de D-flip-flop:



T	D	Q_{n+1}	$\overline{Q}_{n+1}$
L	L	Q_n	$\overline{Q}_n$
L	H	Q_n	$\overline{Q}_n$
H	L	L	H
H	H	H	L

De functietabel is identiek aan die van een latch-flip-flop. Het enige verschil is dat de D-flip-flop een "edge triggered" (flankgestuurde) flip-flop is. Dit wordt aangegeven met de driehoek, binnen het kader, aan de klokingang getekend.

OPMERKING

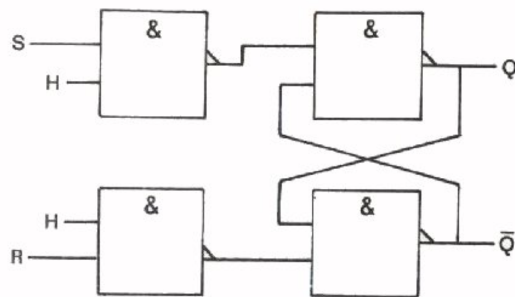
De "C1" en "1D" aanduidingen binnen het kader van een latch- en een D-flip-flop worden in de normen voor logische symbolen gedefinieerd als "afhankelijkheidsnotatie".
De afhankelijkheidsnotatie wordt tot stand gebracht door:

1. de beïnvloedende ingang te voorzien van een speciale letter die de betrokken afhankelijkheid aangeeft, gevolgd door een identificatienummer. In de bovengenoemde voorbeelden is C de speciale letter (C is gereserveerd voor de Commando-afhankelijkheid) en het identificatienummer is hier 1, maar kan ieder geheel getal zijn.

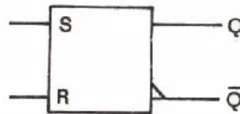
2. elke in- of uitgang die van die beïnvloedende ingang afhankelijk is, te voorzien van hetzelfde nummer. Dit identificatienummer moet vóór het lettersymbool van de in- of uitgang geplaatst worden, bijvoorbeeld 1D. 1D betekent dat ingang D wordt beïnvloed door ingang C1.

SAMENVATTING

- De uit 4 inverterende AND-poorten samengestelde SR-flip-flop.



schakeling

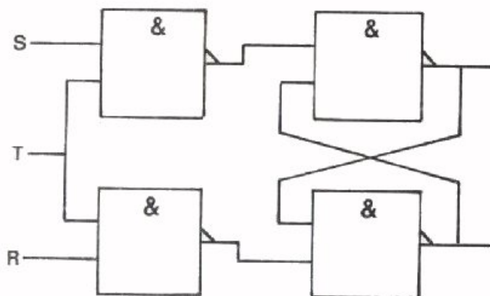


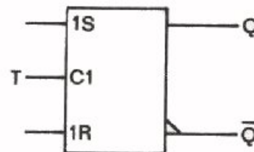
symbool

S	R	Q_{n+1}	$\overline{Q}_{n+i}$
L	L	Q_n	$\overline{Q}_n$
L	H	L	H
H	L	H	L
H	H	H	H

functietabel

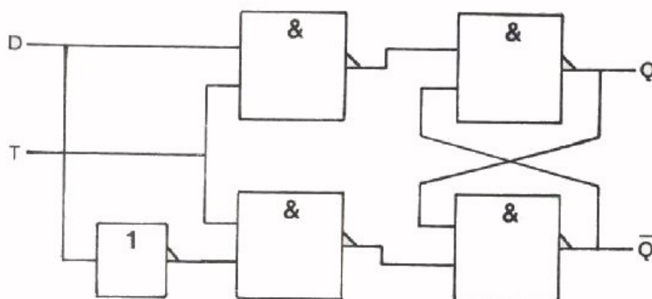
- De uit 4 inverterende AND-poorten samengestelde geklokte SR-flip-flop.

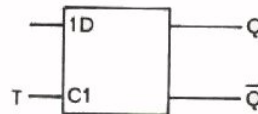




T	S	R	Q_{n+1}	$\overline{Q}_{n+1}$
L	L	L	Q_n	$\overline{Q}_n$
L	L	H	Q_n	$\overline{Q}_n$
L	H	L	Q_n	$\overline{Q}_n$
L	H	H	Q_n	$\overline{Q}_n$
H	L	L	Q_n	$\overline{Q}_n$
H	L	H	L	H
H	H	L	H	L
H	H	H	H	H

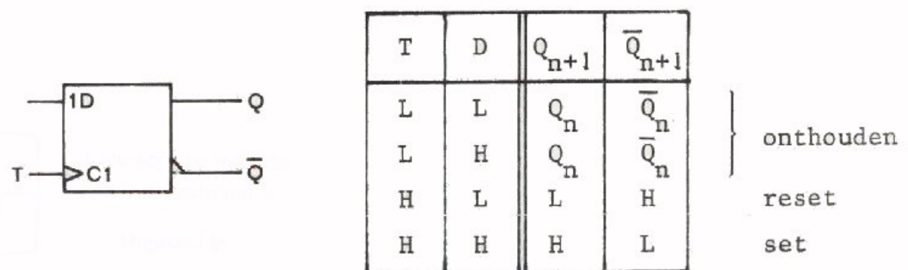
- Een latch-flip-flop (statische klokingang, z.g. "pulse triggered").





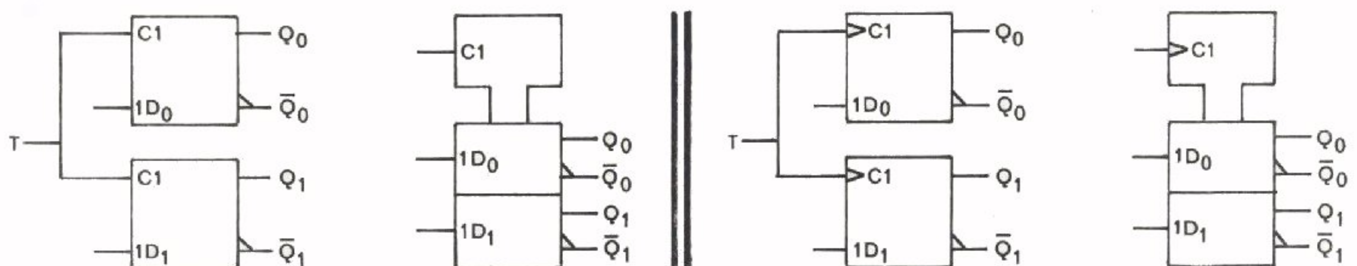
T	D	Q_{n+1}	$\overline{Q}_{n+1}$
L	L	Q_n	$\overline{Q}_n$
L	H	Q_n	$\overline{Q}_n$
H	L	L	H
H	H	H	L

- Een D-flip-flop (dynamische klokingang, z.g. "edge triggered").



T	D	Q_{n+1}	$\bar{Q}_{n+1}$	
L	L	Q_n	$\bar{Q}_n$	onthouden
L	H	Q_n	$\bar{Q}_n$	onthouden
H	L	L	H	reset
H	H	H	L	set

- Uit 2 latch-flip-flops, resp. D-flip-flops, samengesteld REGISTER.



- De 8-LED indicator heeft 8 ingangen B_0 t/m B_7, die elk van 2 doorverbonden aansluitpunten zijn voorzien.
  Sluit men een ingang op 0 V (= L) aan, dan brandt de bijbehorende LED niet.
  Sluit men een ingang op 5 V (= H) aan, dan brandt de bijbehorende LED wel.

- De 8-BIT selector is een module, waarmee men een willekeurige combinatie van 8 bits kan maken door ingangen al of niet aan 0 V te leggen.
  De selector heeft 8 enkele ingangen en 8 dubbel uitgevoerde uitgangen.
  Wordt een ingang met 0 V doorverbonden, dan is de bijbehorende uitgang B hoog (H).
  Wordt een ingang NIET met 0 V doorverbonden, dan is de bijbehorende uitgang B laag (L).

- Een bij voorbeeld uit 4 flip-flops samengesteld register kan een uit 4 bits bestaande informatie tijdelijk (d.w.z., zolang de voedingsspanning eenwezig blijft) onthouden. Zo'n register doet dan dienst als GEHEUGEN.

- Een 4-bits informatie noemt men een "nibble".
  Een 8-bits informatie noemt men een "byte".
  De LSB (= "Least Significant Bit") is de bit met het kleinste gewicht.
  De MSB (= "Most Significant Bit") is de bit met het grootste gewicht.

3 DATA-TRANSPORT

INLEIDING

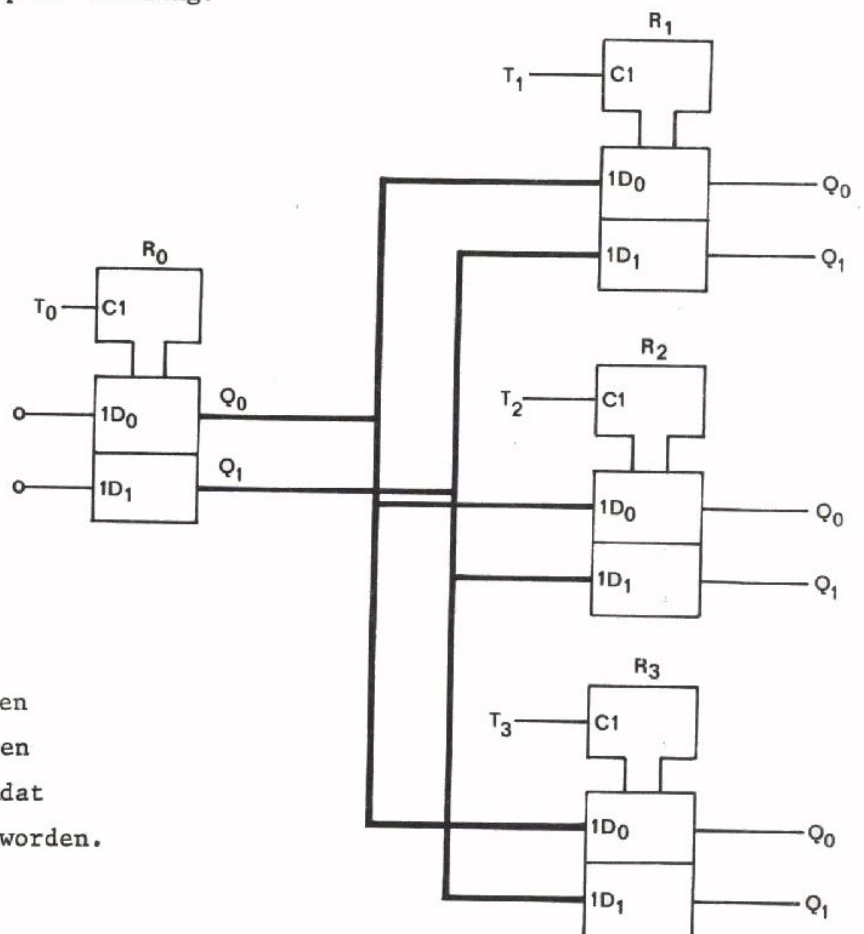
Het gebeurt vaak dat informatie (in het Engels: data) die zich in een register bevindt, van dat register overgebracht moet worden naar één of meer andere registers. Dit transport vindt plaats in een bundel van geleiders die de ingangen en/of de uitgangen van de verschillende registers met elkaar verbindt. Deze bundel noemen we een "bus". De twee belangrijkste bussystemen die in deze les aan de orde komen zijn de "databus" en de "controlebus". De "databus" is het geleider systeem waarover het datatransport plaats vindt. De "controlebus" is het geleider systeem waarover het transport plaats vindt van de commandosignalen die bevelen waarheen de data moet worden overgebracht of waar vandaan de data moet worden opgehaald. Deze begrippen gaan we nu duidelijk maken.

Het begrip "databus" wordt geïllustreerd met onderstaande tekening. Voordat we deze tekening gaan bekijken moeten we even stilstaan bij het gebruik van het engelse woord "data".

Data is het meervoud van "datum". Het laatste woord betekent "gegeven" of "informatie-element". Eén bit zou dus een "datum" moeten zijn, één element van de informatie. Aangezien dat we, in de computerwereld, doorgaans werken met meer dan één bit, wordt uitsluitend het woord "data" gebruikt (dus de meervoudvorm).

In deze cursus zullen we regelmatig engelse woorden gebruiken. Dit kunnen we moeilijk vermijden, want de "computertaal" is engels. Dit zal later blijken wanneer U een microcomputer zult leren programmeren.

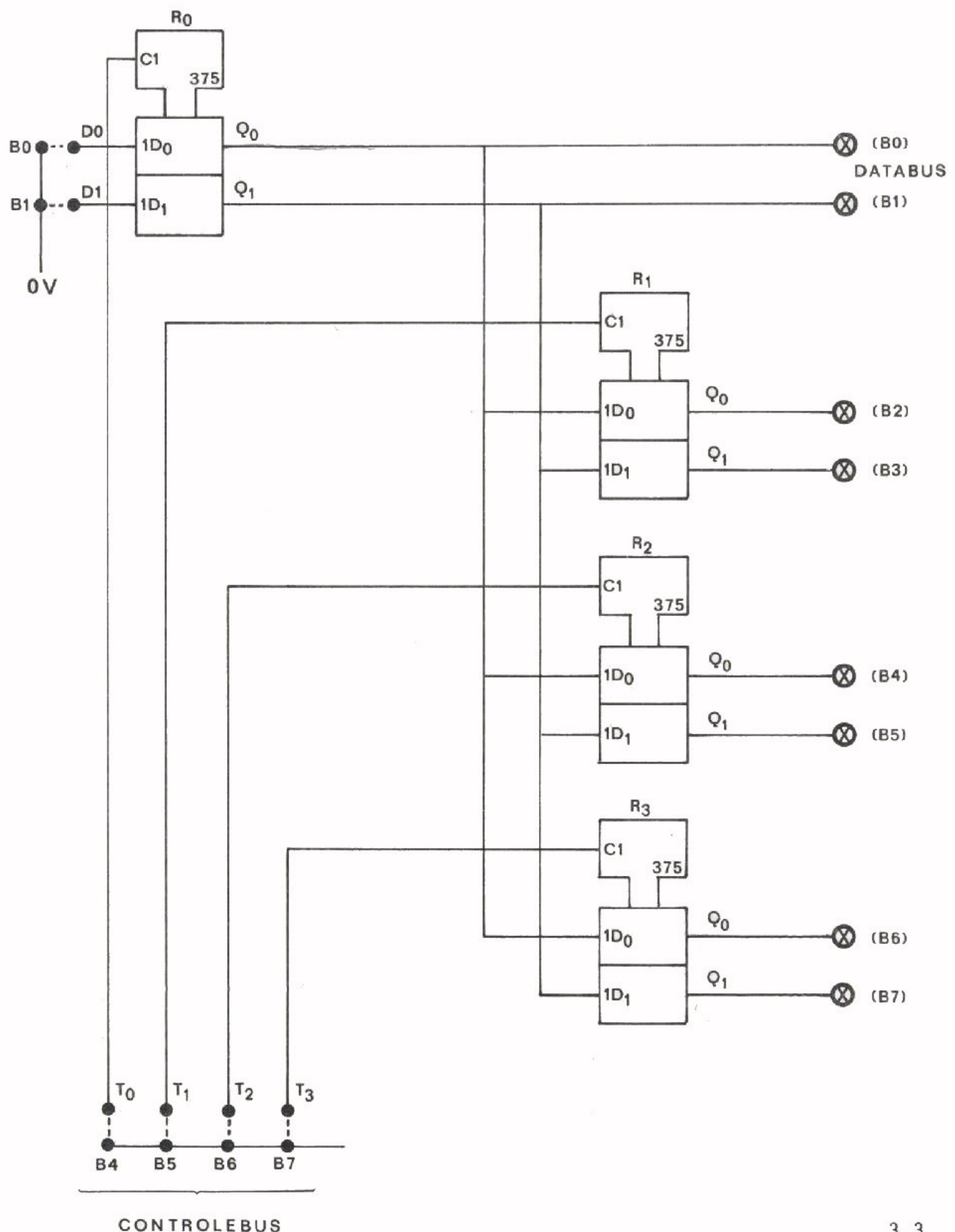
We komen nu terug op de tekening.

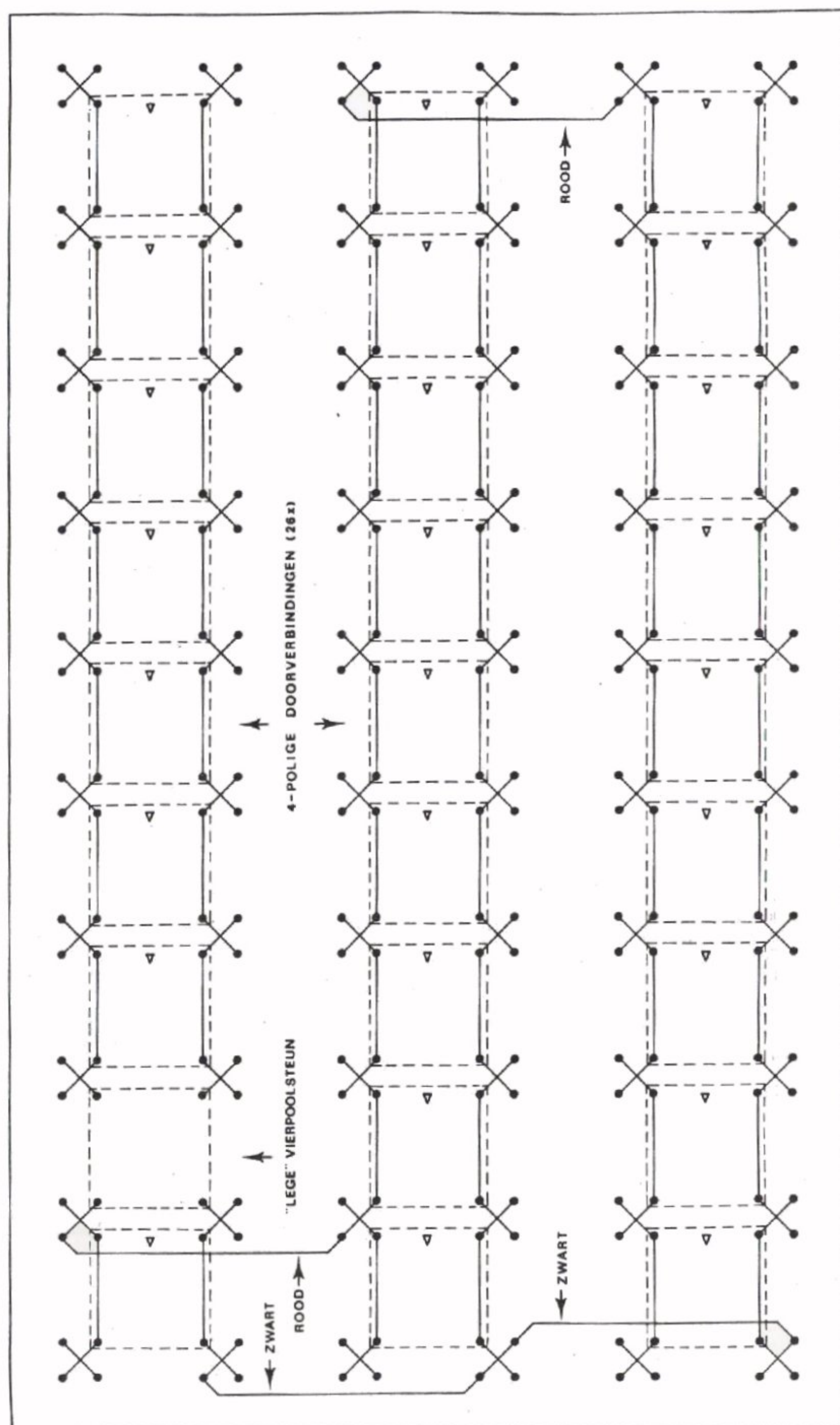


OPMERKING:
In de symbolen hebben we de uitgangen $\overline{Q}_0$ en $\overline{Q}_1$ niet getekend omdat deze niet gebruikt worden.

In bovenstaande tekening gebruiken we vier 2-bits registers. Drie van deze registers (R_1, R_2 en R_3) zijn met hun ingangen parallel geschakeld aan de uitgangen van register R_0. Als de T-ingang van register R_0, H wordt, komen de toestanden van de ingangen $1D_0$ en $1D_1$ van register R_0 beschikbaar aan de uitgangen Q_0 en Q_1 van R_0 en aan de ingangen $1D_0$ en $1D_1$ van R_1, R_2 en R_3. Men zegt dat de data aanwezig zijn op de dik getekende lijnen waarop de registers aangesloten zijn. Deze lijnen vormen de databus. In dit geval hebben we een 2-bits databus.

In de tweede stap willen we de data die op de databus beschikbaar zijn, alleen in R_1 opslaan, en niet in R_2 en R_3. Dit kan door T_1 hoog te maken. en T_2 en T_3 laag te houden. Op deze manier hebben we register R_1 geselecteerd. De selectie gebeurt dus met behulp van commando's die aan de T-ingangen van de registers worden gegeven. In de volgende figuur worden die commando's gegeven met behulp van de 8-BIT selector. De bundel geleiders die de selector met de T-ingangen verbindt noemen we de controlebus. In dit geval hebben we te maken met een 4-bits controlebus. Deze situatie gaan we ervaren in de volgende opdracht.




ROOD
4-POLIGE DOORVERBINDINGEN (26x)
"LEGE" VIERPOOLSTEUN
ROOD
ZWART
ZWART

HET VOORBEREIDEN VAN HET SCHAKELPANEEL

Op de linker pagina ziet U het schakelpaneel getekend waarop een aantal doorverbindingen geplaatst zijn. Deze dienen om o.a. de voedingslijnen (+5 V en 0 V) over het paneel beschikbaar te stellen aan de modulen die we gaan gebruiken. U gaat deze doorverbindingen als volgt aanbrengen:

- Plaats eerst de verticale smalle doorverbindingen zoals aangegeven (twee rode en twee zwarte). De rode doorverbindingen dienen voor de positieve spanning (+ 5 V) en de zwarte voor de 0 V.

- Plaats dan de 26 vierpool doorverbindingen. Zorg ervoor dat het driehoek merk op de doorverbindingen altijd rechts staat. Let er op dat één positie leeg blijft, de tweede positie op de eerste rij!

- De onderdelendoos bevat een ongemonteerde vierpool, een vierpolige plastic houder zonder onderdeel of kortsluitingsdraad. Plaats deze in de tweede positie op de eerste rij. De reden hiervoor zal U later in de cursus duidelijk worden.

- Uw schakelpaneel is nu klaar om de meeste proeven uit te voeren.

OPMERKING

Op de tekening van pagina 3.3, ziet U de aanduidingen "B0" en "(B0)". We spreken nu af dat:

- B0 komt overeen met de aanduiding B0 op de 8-BIT selector.

- (B0) komt overeen met de aanduiding B0 op de 8-LED indicator.

Deze afspraak wordt tijdens de hele cursus aangehouden.

2 BITS DATA BUS

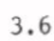
8-LED INDICATOR

74LS375

8-BIT SELECTOR

VOEDINGS-STABILISATOR

Gr = groen
Ge = geel
R = rood
O = oranje
B = blauw



OPDRACHT 3.1: DE WERKING VAN DATA- EN CONTROLEBUSSEN

- Bouw de schakeling van pagina 3.3 zoals hiernaast is weergegeven. Let er op dat U op de bovenste rij, helemaal links de voedingsstabilisator plaatst. De verbindingen tussen de modulen zijn voorzien van de kleur van de te gebruiken snoeren.

- U heeft zeker gemerkt dat het grootste deel van de bovenste rij van het schakelpaneel als "databus" dient. Deze databus wordt van de voedingsstabilisator gescheiden door middel van de "lege" vierpoolsteun.

- Alle doorverbindingen naar de databus worden uitgevoerd door middel van groene snoeren. Deze regel zullen we in alle experimenten toepassen.

- U gaat nu verschillende signalen aan de databus toevoeren via register R_0 en ze in een van de andere registers opslaan. Gebruik hiervoor de volgende tabel.

	Controlebus						R_0		R_1		R_2		R_3	
	$1D_0$ (R_0)	$1D_1$ (R_0)	T_0	T_1	T_2	T_3	Q_0	Q_1	Q_0	Q_1	Q_0	Q_1	Q_0	Q_1
1	L	L	H	H	H	H								
2	L	L	H	L	L	L								
3	H	L	H	L	L	L								
4	H	L	L	L	L	L								
5	L	L	L	H	L	L								
6	L	L	L	L	L	L								
7	L	H	H	L	L	L								
8	L	H	L	L	L	L								
9	L	L	L	L	H	L								
10	L	L	L	L	L	L								
11	H	H	H	L	L	L								
12	H	H	L	L	L	L								
13	L	L	L	L	L	H								

- Vul voor iedere regel de Q-waarden in.

- Breek de schakeling af.

RESULTATEN

De handelingen van de eerste regel zijn bedoeld om al de registers te resetten (al de uitgangen worden L). In de tweede regel selecteren we register R_0 en in de derde regel bieden we $1D_0$ = H en $1D_1$ = L aan, aan de ingangen van R_0. Deze waarden worden door Q_0 en Q_1 overgenomen en op de databus geplaatst. Als we nu de T-ingang van R_0 L maken, blijft R_0 het signaal Q_0 = H en Q_1 = L onthouden. Deze informatie blijft dus beschikbaar op de databus. Schakelen we nu T_1 op H, dan wordt de informatie ook overgenomen door de Q-uitgangen van R_1 (regel 5).
Probeer zelf de volgende regels te verklaren op de manier als hierboven is voorgedaan.

CONCLUSIES

In de schakeling van de laatste opdracht hebben we gezien dat de informatie die aangeboden is aan de ingangen van register R_0, op de databus gezet wordt door de klokingang T_0 hoog te maken. Elk register dat met zijn ingangen met de bus is verbonden kan deze informatie overnemen als het "geselecteerd" wordt. Registers worden geselecteerd om informatie over te nemen door een H-signaal aan te brengen op de T-ingangen die op de controlebus aangesloten zijn.
De databus zoals hierboven beschreven heet een "UNI DIRECTIONAL" bus-systeem, een eenrichtingsverkeer van DATA. In een dergelijk systeem kan EEN register informatie op de bus zetten terwijl de andere registers de informatie van de bus kunnen opslaan.

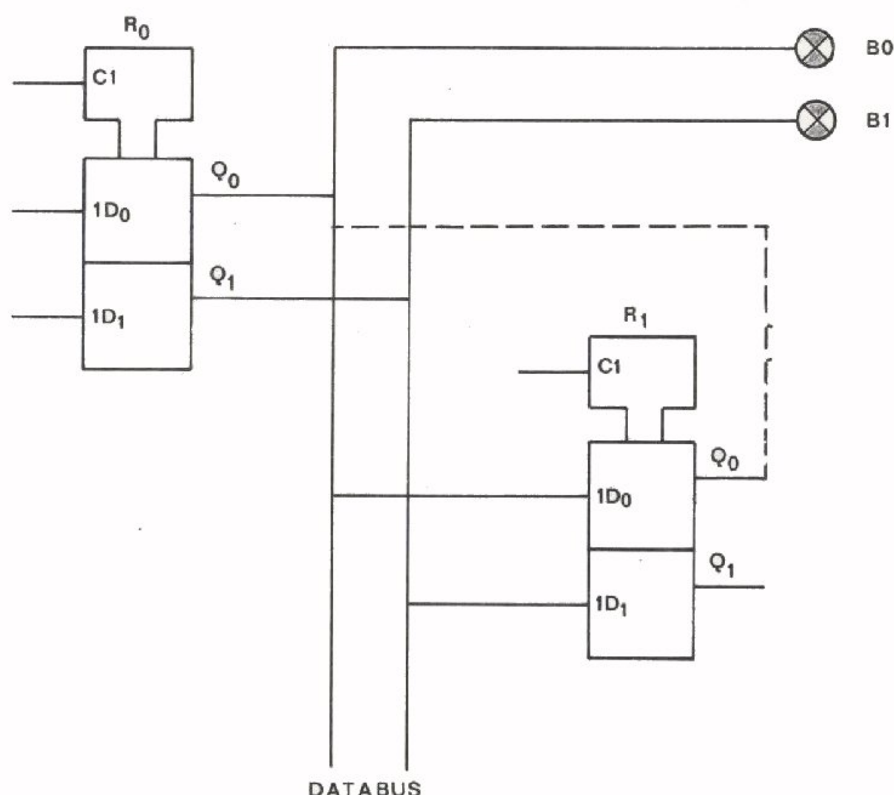
OPMERKING

De schakeling van de vorige opdracht doet vermoeden dat de controlebus altijd evenveel geleiders bevat als het aantal gebruikte registers. Dit is echter niet zo, hetgeen later verklaard zal worden.

BI-DIRECTIONAL DATABUS

Bussystemen via welke verscheidene registers om de beurt kunnen "schrijven" en "lezen", heten "BI-DIRECTIONAL" databussystemen. De data stroom gaat dan over dezelfde verbindingen, dán weer in de ene, dán weer in de andere richting. Natuurlijk moeten er in een bi-directional databussysteem speciale voorzorgsmaatregelen worden genomen. Stel dat in het circuit van pagina 3.3 op een bepaald moment de inhoud van register R_1 weer op de databus moet verschijnen. De Q-uitgangen van register R_1 kunnen dan niet zo maar rechtstreeks op de databus aangesloten worden.

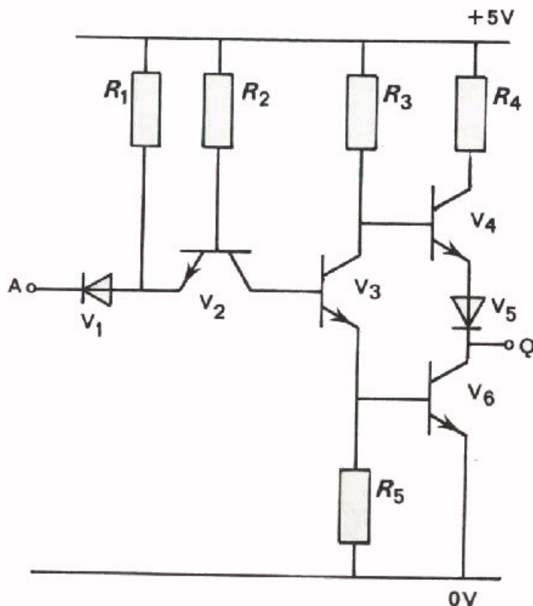
Bekijk de volgende situatie.

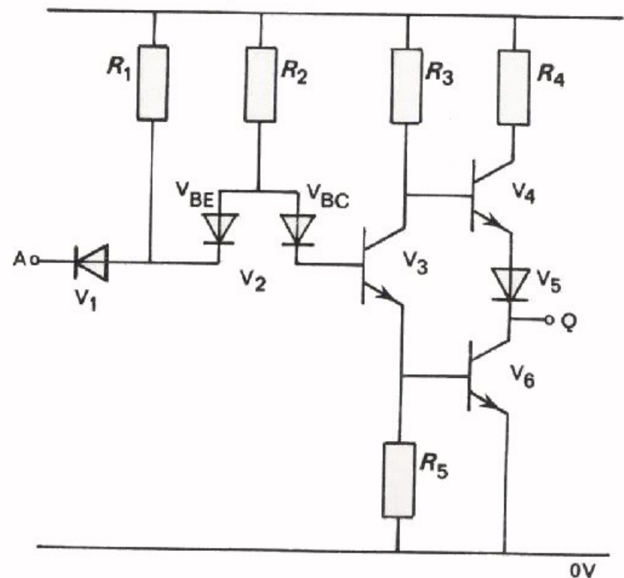


Wat gaat er nu gebeuren als we uitgang Q_0 van register R_1 rechtstreeks met de databus verbinden (stippellijn)? We nemen daarbij aan dat C1 van register R_1 op 0 V staat, zodat de ingang $1D_0$ en de uitgang Q_0 van R_1 niet kortgesloten zijn. Om te begrijpen wat er dan gaat gebeuren, is het nuttig om eerst het principe van een TTL poort nog eens nader te bekijken.

PRINCIPE VAN TTL

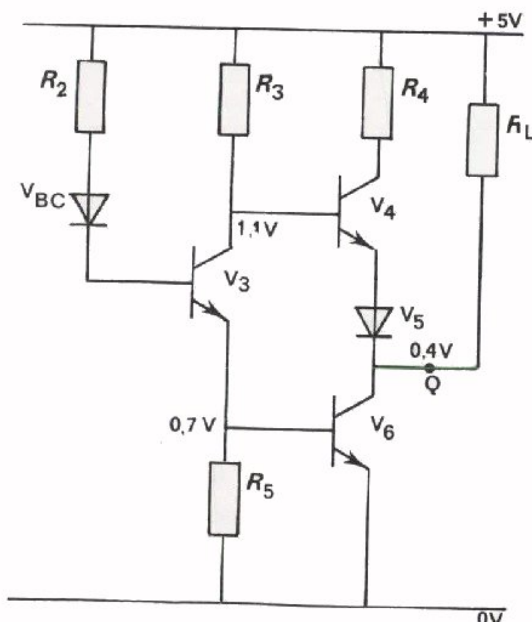
In deel D hebben we het principe van een inverterende AND-poort in TTL-uitvoering uitgelegd. Deze gaan we echter nog eens bekijken met behulp van onderstaande tekeningen.





De linker figuur geeft het principe schema van een inverterende AND-poort. In feite is transistor V_2 een multi-emitter transistor. We hebben echter één ingang getekend (A) om de verklaring te vereenvoudigen. Transistor V_2 bevat twee dioden, de B-E diode en de B-C diode. Immers, zowel tussen B en E als tussen B en C bestaat een PN-overgang.

Ligt ingang A aan de + 5 V, dan zijn de dioden V_1 en V_{BE} gesperd. Er loopt een stroom door R_2 en V_{BC} (zie rechter schema boven). Transistor V_3 is dan in geleiding en gaat in verzadiging. Zijn collector-emitter spanning is dan ongeveer U_{CE3} = 0,4 V. De emitter spanning U_{E3} kan nooit hoger worden dan 0,7 V want aan R_5 staat de B-E diode van transistor V_6 parallel en deze diode is geleidend.



De collectorspanning U_{C3} bedraagt dus:

$$U_{C3} = U_{E3} + U_{CE3} = U_{BE6} + U_{CE3}$$

$$= 0{,}7 + 0{,}4 = 1{,}1 \text{ V.}$$

In nevenstaande tekening hebben we een belasting weerstand R_L toegevoegd.

Deze weerstand simuleert de ingangsimpedantie van de poorten die aan de inverterende AND aangesloten zijn. Transistor V_6 is geleidend: er loopt een stroom door R_L en V_6.
Transistor V_4 is gesperd, want:

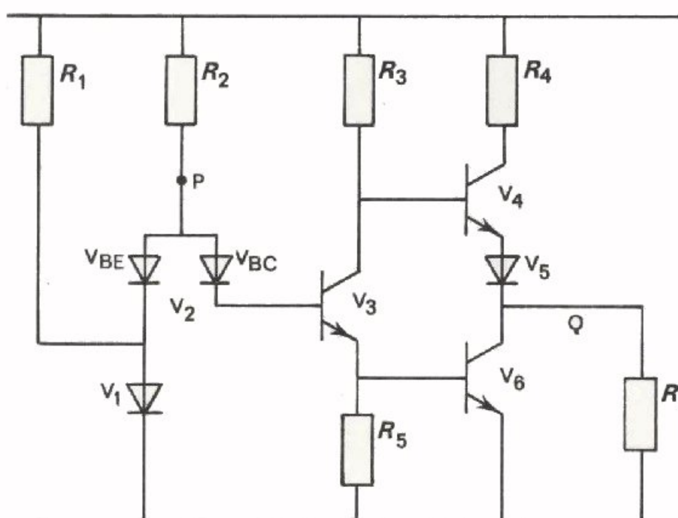
$$U_{BE4} + U_{AK5} + U_{CE6} = 1{,}1 \text{ V}.$$

$U_{C6} \simeq 0{,}4$ V, zodat $U_{BE4} + U_{AK5} \simeq 0{,}7$ V.
Deze spanning is onvoldoende om deze in serie staande dioden te doen geleiden.

$$U_{C6} = U_Q \simeq 0{,}4 \text{ V, zodat } Q = L.$$

De uitgangsimpedantie van de inverterende AND-poort is gelijk aan de uitgangsimpedantie van een transistor in de verzadigingstoestand; deze is dus zeer laag, max. enkele ohm.

Wat gebeurt er nu als ingang A is aangesloten op de 0 V? Dioden V_1 en V_{BE} geleiden.



$$U_P = U_{BE} + U_{V1}$$
$$= 0{,}7 + 0{,}7 = 1{,}4 \text{ V}.$$

Deze spanning verdeelt zich:

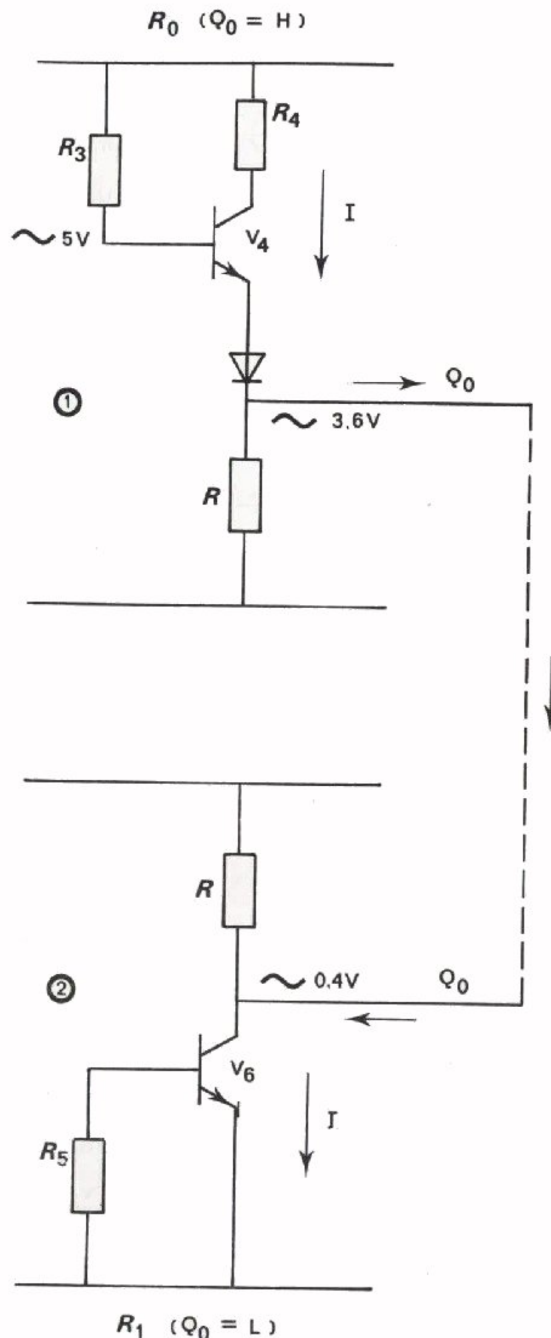
$$U_P = U_{BC} + U_{BE3} + U_{BE6}$$

en dit is niet voldoende om de drie in serie staande dioden te doen geleiden. Dus V_3 en V_6 zijn gesperd. De collector van V_3 ligt ongeveer op 5 V, zodat V_4 geleidt. Zijn uitgangsimpedantie is laag, want V_4 werkt hier als emittervolger. Verder geldt:

$$U_Q \simeq 5 \text{ V} - U_{BE4} - U_{V5} \simeq 3{,}6 \text{ V}.$$

Dus Q = H.

We hebben gezien dat de uitgangsimpedantie van een TTL-poort zoals hier beschreven zeer laag is, ongeacht de toestand van de uitgang. Niet alle TTL-poorten hebben een uitgangscircuit zoals hier beschreven. De uitgangsimpedantie is echter altijd laag, ongeacht de uitgangstoestand.
Op de volgende pagina hebben we de toestand van pagina 3.5 weergegeven. We nemen aan dat de uitgangscircuits van de registers van hetzelfde type zijn als die van de zo juist beschreven inverterende AND-poort.



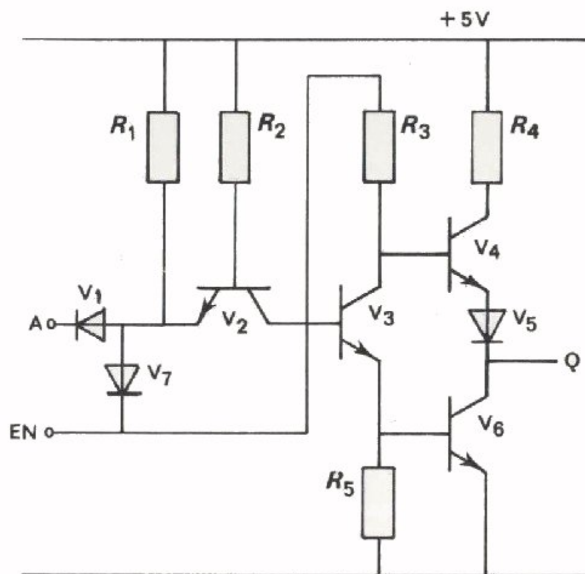
Als de inverterende AND-poort ① een uitgang H heeft, en de inverterende AND-poort ② een uitgang L, en deze uitgangen worden vervolgens doorverbonden, dan zal er een grote kortsluitstroom optreden waardoor componenten defect kunnen raken. Vergeet niet dat de uitgangsweerstand van elk van de TTL-poorten zeer laag is: vandaar de uitdrukking "kortsluitstroom".

CONCLUSIE

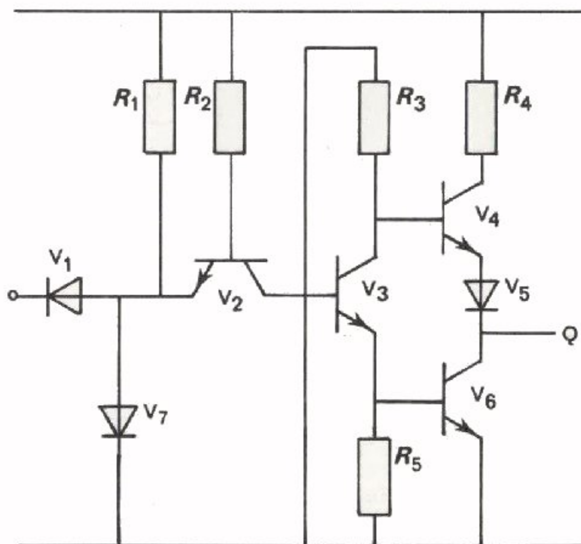
Als er een register op de databus schrijft, moeten de uitgangen van de andere registers die op de bus aangesloten zijn afgeschakeld worden. Hiervoor zijn speciale IC's ontwikkeld waarvan de uitgangen in de zogenaamde "hoogohmige toestand" gezet kunnen worden via een speciale ingang.

DE HOOGOHMIGE TOESTAND

Het circuit hieronder is de inverterende AND-poort in TTL-uitvoering die we eerder hebben bekeken. De collectorweerstand van V_3 is echter niet meer rechtstreeks aan de + 5 V aangesloten, maar apart uitgevoerd via de ingang EN.



Als ingang EN op + 5 V wordt aangesloten, is de situatie zoals we eerder hebben besproken. Diode V_7 is immers gesperd. De situatie is echter anders als EN met de 0 V wordt doorverbonden. In dat geval is U_{C3} = 0 en zijn V_4 en V_6 gesperd, zodat hun uitgangsimpedantie zeer hoog is. Diode V_7 is aangebracht om te vermijden dat als A met + 5 V is verbonden, er een stroom zou kunnen lopen door R_2, V_{BC2}, V_{BE3} en V_{BE6} die V_6 uit de gesperde toestand zou brengen. Verklaar dit zelf aan de hand van nevenstaande schema.

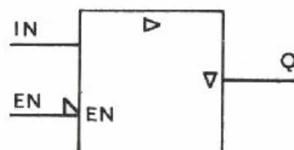


CONCLUSIE

De EN-ingang van een IC met hoogohmige toestand wordt de "enable" ingang genoemd. De enable ingang controleert, afhankelijk van het niveau waarop deze ingang staat, de uitgang. Als, in dit geval, de EN-ingang L is, geeft de IC niet meer het signaal door zoals dat in de poort of de flipflop aanwezig is; de IC heeft dan een hoge uitgangsimpedantie waardoor hij geen invloed meer uitoefent op de databus.

DE HEX BUFFER MODULE MET HOOGOHMIGE TOESTAND

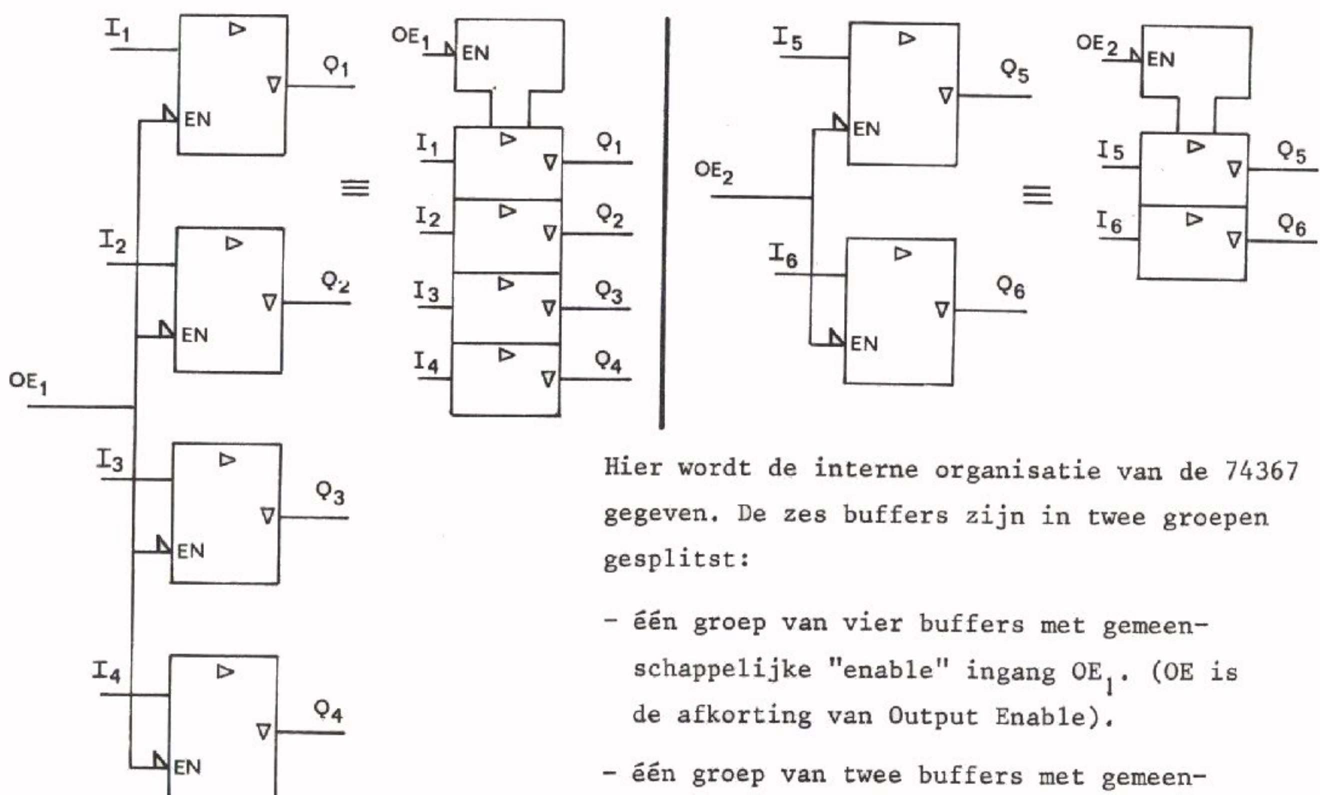
Een register wordt in een bussysteem altijd gebruikt in combinatie met een "hoogohmige toestand uitgang". In tegenstelling tot het besproken vereenvoudigde principe circuit van een TTL-poort met hoogohmige toestand, zijn de meeste IC's die met hoogohmige toestand uitgevoerd zijn actief indien de enable ingang laag is; de hoge impedantie uitgang wordt dan bereikt als de enable ingang hoog is. Wij beschikken over een Hex buffer-module met hoogohmige toestand: de 74367. Hieronder vindt U het grafische symbool van deze buffer en de functietabel.



EN	IN	Q
L	L	L
L	H	H
H	L	Z
H	H	Z

De driehoek ▽ in het symbool geeft aan dat de uitgang de hoogohmige toestand kan aannemen.

Wanneer de letter Z gebruikt wordt in een functie- of in een volgordetabel, betekent het "hoogohmige toestand".



Hier wordt de interne organisatie van de 74367 gegeven. De zes buffers zijn in twee groepen gesplitst:

- één groep van vier buffers met gemeenschappelijke "enable" ingang OE_1. (OE is de afkorting van Output Enable).
- één groep van twee buffers met gemeenschappelijke "enable" ingang OE_2.

Voor elke groep hebben we ernaast het desbetreffende logische symbool getekend.

OPMERKING

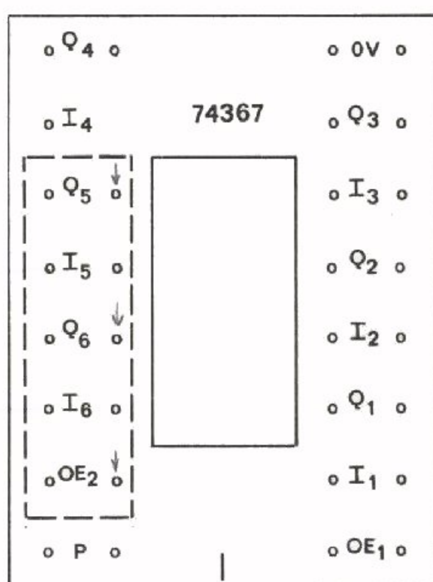
Een buffer (of versterker) met hoogohmige toestand wordt in de litterature vaak beschreven als een "Three-states buffer". Dit betekent dat de versterker drie uitgangstoestanden ("states") kan hebben:

a. De toestand met hoogohmige uitgang.

b. De toestand waarbij het uitgangsniveau laag is.

c. De toestand waarbij het uitgangsniveau hoog is.

Mogelijkheid a wordt ook de "derde toestand" van de versterker genoemd. In deze cursus hebben we gekozen voor de "hoogohmige toestand" benaming, die rechtstreeks de eigenschap van de uitgang verklaart.



Hiernaast vindt U de tekening van de bovenplaat van de module. De stippellijnen duiden aan dat OE_2 betrekking heeft op de buffers 5 en 6.

OPDRACHT 3.2: METINGEN AAN DE HEX BUFFER-MODULE (A)

- Prik de 8-BIT selector, de 74367-module en de 8-LED indicator naast elkaar op het schakelpaneel.

- Verbind ingang I_1 van de 74367 met B1 van de 8-BIT selector en uitgang Q_1 met LED B1.

- Vul de volgende functietabel is:

OE_1	I_1	Q_1
L	L	
L	H	
H	L	
H	H	

- Verbind ingang I_2 met B2 van de 8-BIT selector en uitgang Q_2 met LED B2.

- Vul nu de volgende functietabel in:

OE_1	I_2	Q_2
L	L	
L	H	
H	L	
H	H	

- Herhaal, zonder de vorige aansluitingen af te breken, de procedure voor buffer 3 en daarna voor buffer 4.

- Breek de schakeling nog niet af.

CONCLUSIES

1. De vier buffers zijn actief wanneer OE, de enable ingang, L is. In dit geval volgt de uitgang de waarde van de bijbehorende ingang.

2. Als OE_1 aan de +5 V (P) aangesloten is, blijven de LED's uit. Dit zou betekenen dat de Q-uitgangen L zijn. Dat dit niet het geval is, zal uit het volgende experiment blijken.

OPDRACHT 3.3: METINGEN AAN DE HEX BUFFER-MODULE (B)

- Breid de schakeling uit met behulp van de 13-segment LED indicator, die U naast de 8-LED indicator plaatst.

- Verbind LED's B1 t/m B4 met respectievelijk de ingangen A t/m D van de 13-segment module.

- Zorg dat OE_1 met 0V verbonden is.

- Zet op de 8-BIT selector B1 = H

  B2 = L

  B3 = L

  B4 = H

  Het getal 9 (1001) is te zien op de 13-segment indicator.

- Sluit nu OE_1 aan P (+5 V).

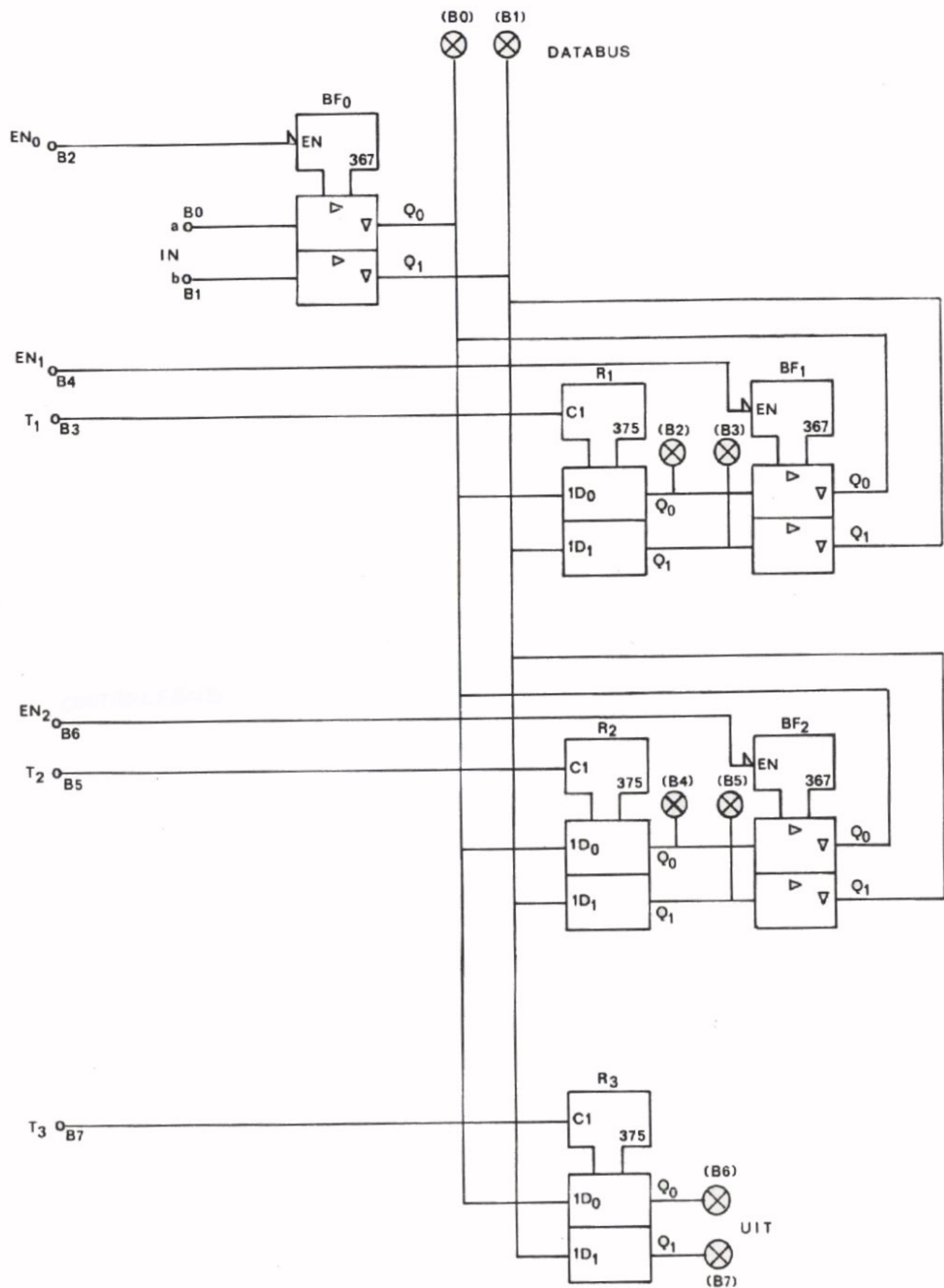
- Constateer dat de LED's van de 8-LED indicator uitgaan. Hieruit zou U kunnen concluderen dat op de Q-uitgangen het getal 0000 aanwezig is.

- De 13-segment module geeft echter het hexadecimaal getal F, dus 1111, aan.

- Maak nu de snoeren van ingangen A t/m D van de 13-segment module los. De uitlezing blijft F.

CONCLUSIE

Wanneer OE_1 aan de +5 V aangesloten is, hebben de Q uitgangen geen invloed op de schakelingen die aangesloten zijn. Als door de 8-LED indicator en de 13-segment module tegelijkertijd 0000 en 1111 worden weergegeven, duidt dit op een hoogohmige toestand.

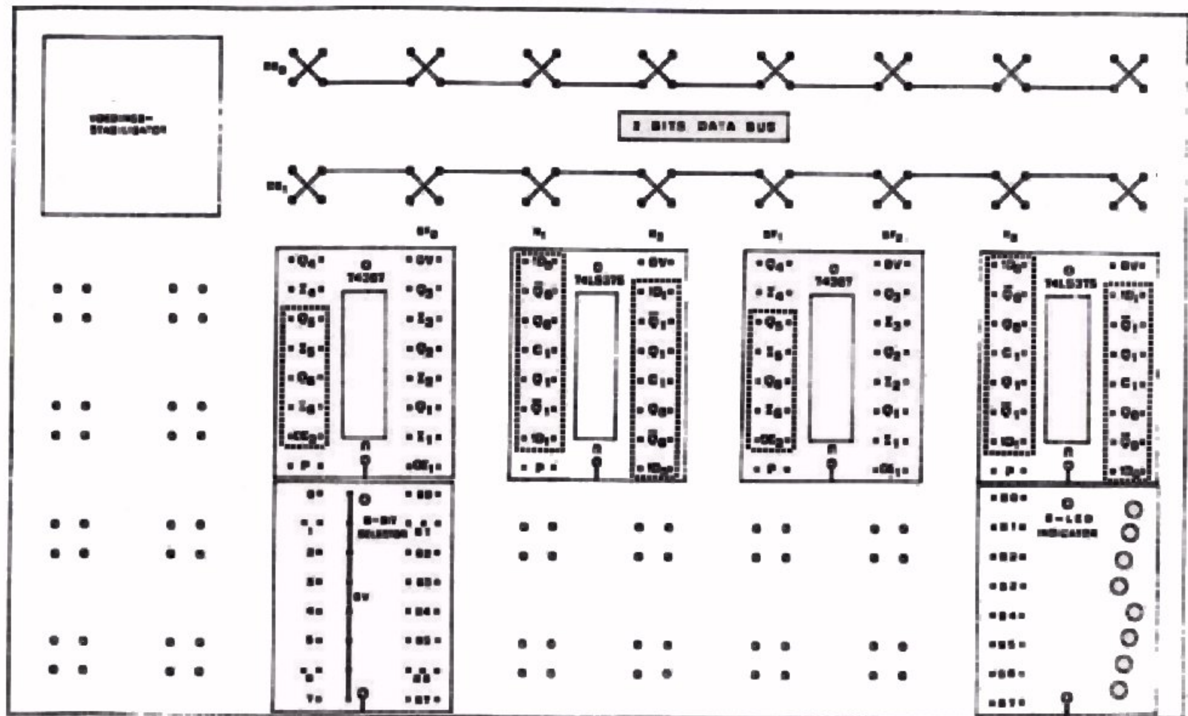

(B0)
(B1)
DATABUS
BF0
EN0
B2
EN
367
B0
a
Q0
IN
Q1
b
B1
EN1
B4
R1
BF1
T1
B3
C1
375
(B2)
(B3)
EN
367
1D0
Q0
Q0
1D1
Q1
Q1
EN2
B6
R2
BF2
T2
B5
C1
375
(B4)
(B5)
EN
367
1D0
Q0
Q0
1D1
Q1
Q1
R3
T3
B7
C1
375
1D0
Q0
(B6)
UIT
1D1
Q1
(B7)

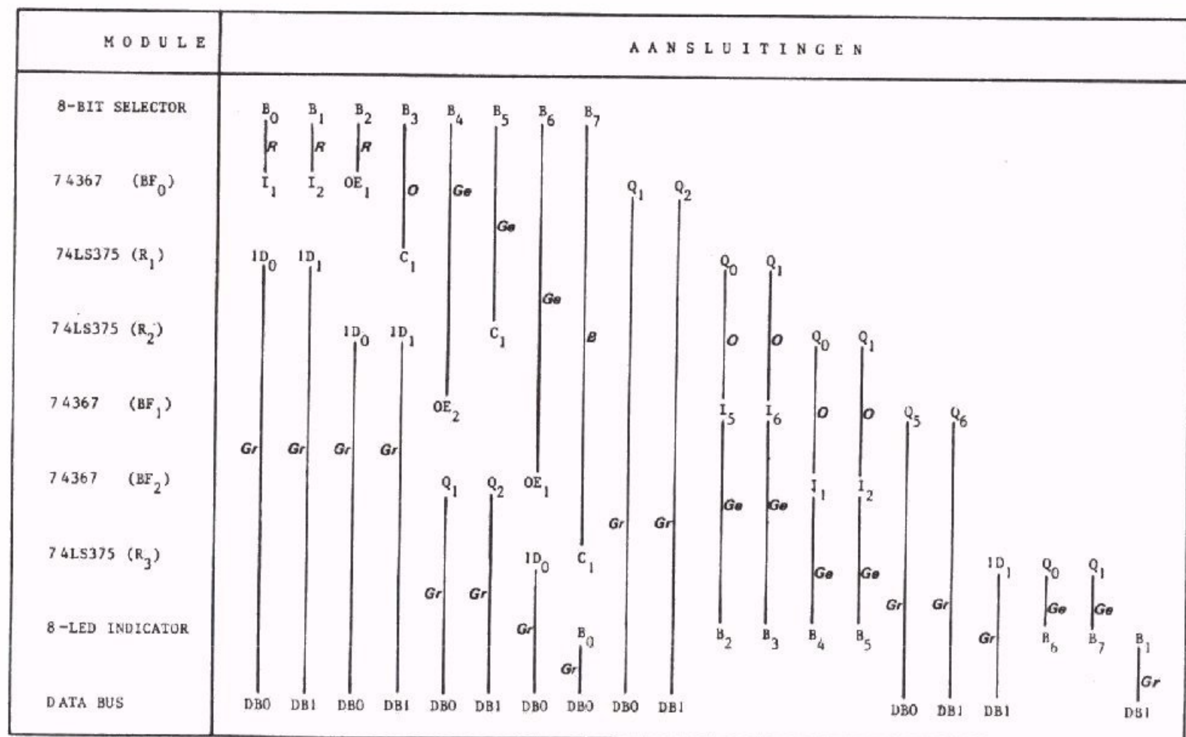
BI-DIRECTIONAL DATABUS

We gaan het circuit van opdracht 3.1 zodaning uitbreiden dat vanuit de registers R_1 en R_2 op de databus geschreven of gelezen kan worden. Het circuit van de vorige pagina laat deze uitbreiding zien. De uitgangen van registers R_1 en R_2 zijn aangesloten op de databus via twee buffers met hoogohmige toestand. Verder wordt register R_0 van het circuit van pagina 3.3 vervangen door een derde buffer met hoogohmige toestand, zodat we van buitenaf (b.v. met behulp van schakelaars) een 2-BITS informatie op de databus kunnen schrijven.

Register R_3 is niet teruggevoerd naar de databus. Dit register vormt de uitgang van het systeem. De standen van deze uitgang worden zichtbaar gemaakt door twee LED's.

In het volgende experiment gebruiken we een 8-BIT selector voor de data en controle-ingangen. De aanduidingen B0 t/m B7 op de tekening geven de desbetreffende uitgangen van deze module aan. Voor de 8-LED indicator zijn de ingangen aangeduid met (B0) t/m (B7).


2 BITS DATA BUS
74367
74LS375
74367
74LS375
8-BIT SELECTOR
8-LED INDICATOR


MODULE
AANSLUITINGEN
8-BIT SELECTOR
74367 (BF0)
74LS375 (R1)
74LS375 (R2)
74367 (BF1)
74367 (BF2)
74LS375 (R3)
8-LED INDICATOR
DATA BUS

- In de vorige opdracht had U de beschikking over een tekening van het schakelpaneel waarop alle verbindingssnoeren getekend waren en ook de kleur van de te gebruiken snoeren gegeven was. Als het aantal doorverbindingen zal groeien met het aantal modulen, zullen dergelijke tekeningen onoverzichtelijk worden. Om dit te vermijden hebben we een methode ontwikkeld die U hiernaast kunt vinden. Op het schakelpaneel zijn de posities van de modulen aangegeven. Daaronder vindt U het bedradingsschema met de aanduiding van de kleur van de te gebruiken snoeren.

  We raden U sterk aan het circuit op te bouwen met behulp van de tekening van pagina 3.18 en daarbij het bedradingsschema aan te houden. Dit heeft twee voordelen:

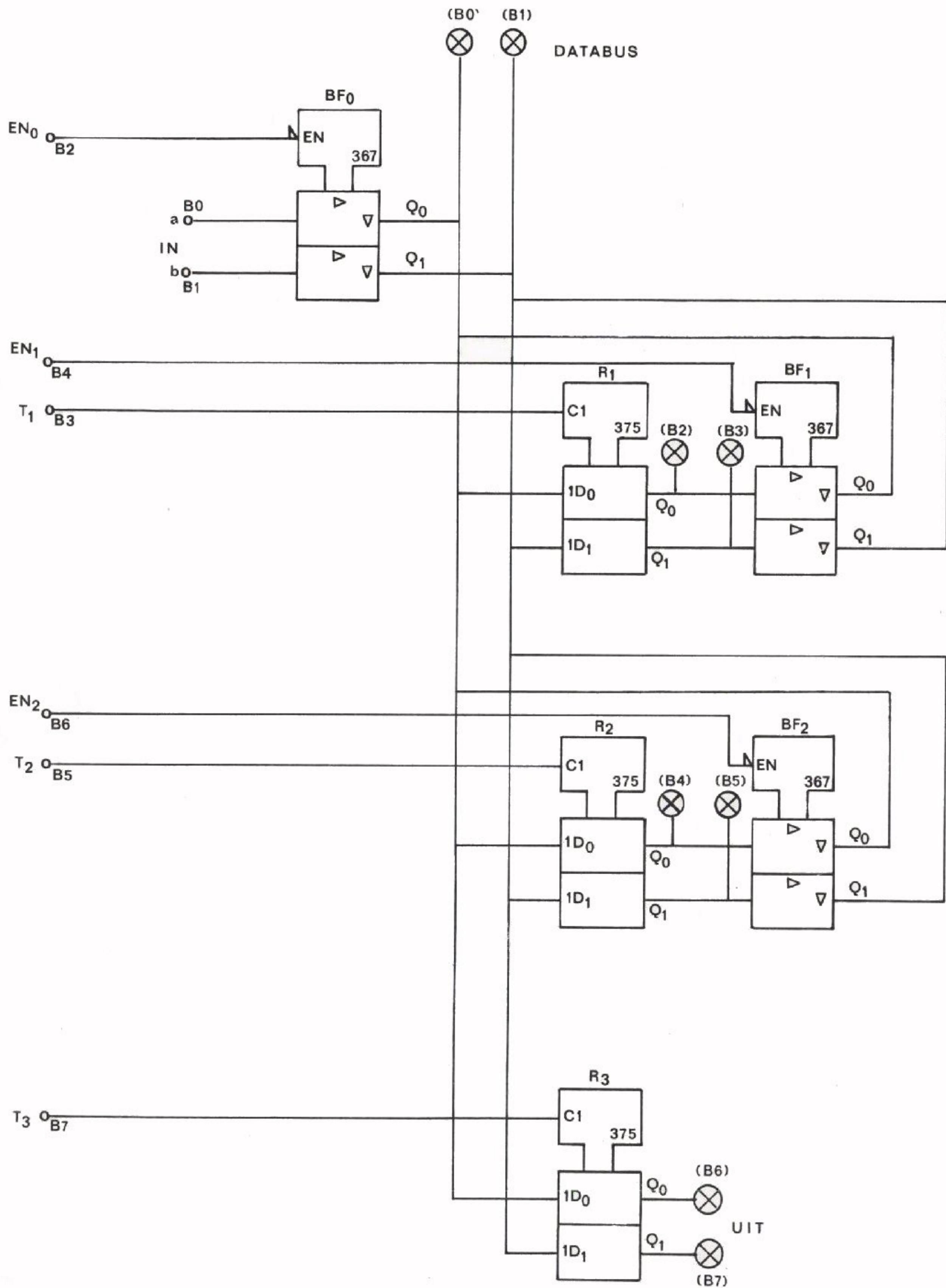
  1. Indien het circuit niet goed functioneert is het mogelijk om snel de bedrading te controleren.
  2. Voor de volgende experimenten behoeft U slechts weinig te veranderen indien U de modulen plaatst zoals is aangegeven.

OPDRACHT 3.4: DE BIDIRECTIONAL DATABUS

- Zorg ervoor dat de EN ingangen op 5 V staan; dus dat er een kortsluitbeugel aanwezig is in de posities B2, B4 en B6 van de 8-BIT selector. De andere posities (B0, B1, B3, B5 en B7) blijven open.

- Op de volgende twee pagina's vindt U links een herhaling van het schema van pagina 3.18 en rechts een tabel die U proefsgewijze moet invullen. Alvorens dit te doen dient U de vier volgende punten goed te lezen.

  1. Bij het inschakelen van de voedingsspanning geven de uitgangen van de registers een willekeurige waarde weer. Deze waarden hebben geen betekenis en zijn vervangen door kruisjes.
  2. Regel 1 van de tabel geeft de startpositie aan zoals hierboven beschreven is.
  3. De tabel gaat tot regel 20, maar U kunt verder experimenteren en Uw resultaten opschrijven in de daarvoor beschikbare ruimte.
  4. Breek daarna de schakeling niet af.


(B0)
(B1)
DATABUS
BF0
EN0
B2
EN
367
B0
a
IN
b
B1
Q0
Q1
EN1
B4
T1
B3
R1
C1
375
(B2)
(B3)
BF1
EN
367
1D0
Q0
1D1
Q1
Q0
Q1
EN2
B6
T2
B5
R2
C1
375
(B4)
(B5)
BF2
EN
367
1D0
Q0
1D1
Q1
Q0
Q1
R3
T3
B7
C1
375
1D0
Q0
(B6)
UIT
1D1
Q1
(B7)

Positie op 8-BIT selector en 8-LED indicator

INGANG		CONTROLES						UITGANGEN								
a	b	EN_o	T_1	EN_1	T_2	EN_2	T_3	DATABUS		R_1		R_2		R_3		
B0	B1	B2	B3	B4	B5	B6	B7	(B0)	(B1)	(B2)	(B3)	(B4)	(B5)	(B6)	(B7)	
L	L	H	L	H	L	H	L			X	X	X	X	X	X	1
H	L	H	L	H	L	H	L			X	X	X	X	X	X	2
H	L	L	L	H	L	H	L			X	X	X	X	X	X	3
H	L	L	H	H	L	H	L					X	X	X	X	4
H	L	L	L	H	L	H	L					X	X	X	X	5
H	L	H	L	H	L	H	L					X	X	X	X	6
L	H	H	L	H	L	H	L					X	X	X	X	7
L	H	L	L	H	L	H	L					X	X	X	X	8
L	H	L	L	H	H	H	L							X	X	9
L	H	L	L	H	L	H	L							X	X	10
L	H	H	L	H	L	H	L							X	X	11
H	H	H	L	H	L	H	L							X	X	12
H	H	L	L	H	L	H	L							X	X	13
H	H	L	L	H	L	H	H									14
H	H	L	L	H	L	H	L									15
H	H	H	L	H	L	H	L									16
H	H	H	L	L	L	H	L									17
H	H	H	L	L	L	H	H									18
H	H	H	L	L	L	H	L									19
H	H	H	L	H	L	H	L									20

WAT HEBBEN WE GEDAAN?

1. In de regels 1 t/m 6 hebben we een getal "01" (LH) in register R_1 opgeslagen.

2. In de regels 7 t/m 11 hebben we een ander getal "10" (HL) in register R_2 opgeslagen.

3. In de regels 12 t/m 16 wordt het getal "11" (HH), via de databus, rechtstreeks naar de uitgang gebracht via register R_3.

4. In de regels 17 t/m 20 hebben we het getal dat in register R_1 was opgeslagen naar de uitgang gebracht (dus opgeslagen in register R_3).

U heeft zeker andere verschuivingen van het ingangssignaal geprobeerd. U kunt ze hieronder in het kort uitleggen.

OPMERKING

De uitlezing van (B0) en (B1) op de 8-LED indicator - dus de data aanwezig op de databus - op de regels 1, 2, 6, 7, 11, 12, 16 en 20, waren altijd "LL". In feite zou U daar de aanduiding "ZZ" moeten inschrijven. Dit kunt U bekijken door een 13-segment indicator naast de 8-LED indicator te prikken. Verbind dan (B0) met A en (B1) met B. Sluit ook C en D aan + 5 V (P). In de bovengenoemde regels zal de 13-segment indicator altijd de letter F aangeven.

CONCLUSIES

- In de laatste opdracht heeft U gezien op welke manier de ingangsinformatie van een register naar een ander via de databus kan verschuiven. Hoe dit gebeuren kan heeft U zelf beslist door, op het juiste moment, een commando te geven aan de desbetreffende controle-ingang.
- Buffer ⓪ dient om het door U gekozen binaire getal op de databus te zetten. Deze buffer wordt, in computertaal, een INGANGSPOORT genoemd. Er wordt ook gezegd dat deze ingangspoort de "buitenwereld" verbindt aan de databus.
- Register R_3 dient om een getal dat op de databus aanwezig is beschikbaar te stellen aan de "buitenwereld". Een dergelijk register noemen we een UITGANGSPOORT.

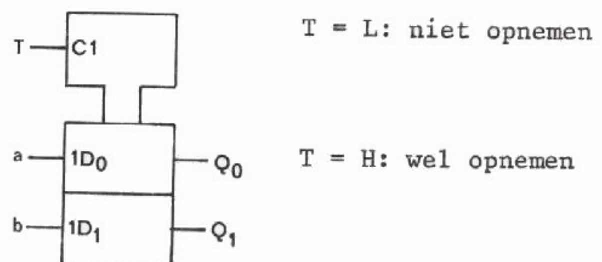
SAMENVATTING

- Informatie of gegevens duidt men aan met de engelse uitdrukking DATA.

- Verbindingen via welke men data overbrengt noemt men DATALIJNEN.
  De gezamenlijke datalijnen noemt men de DATABUS.

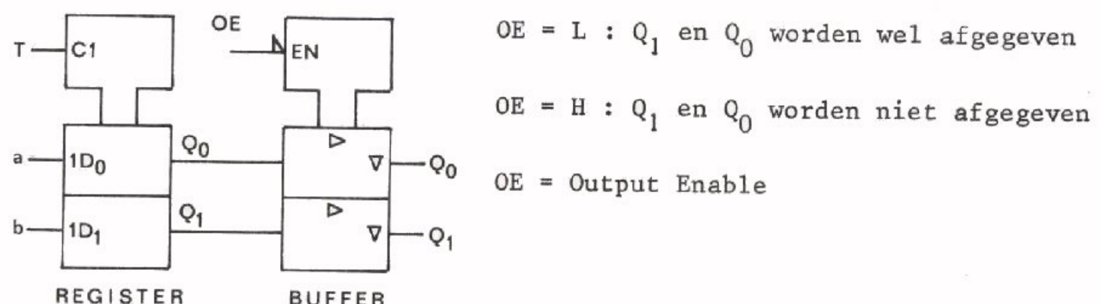
- Verbindingen via welke commandosignalen worden verstuurd die bevelen waar de data naartoe moeten gaan en waarvandaan de data opgehaald moeten worden, noemt men CONTROLELIJNEN.
  De gezamenlijke controlelijnen noemt men de CONTROLEBUS

- Het kloksignaal T commandeert of een register de aangeboden informatie (a en b) al of niet OPNEEMT.



- Twee of meer registers mogen niet tegelijkertijd hun uitgangssignalen aan dezelfde datalijnen toevoeren. Dit omdat er in geval van tegengestelde uitgangssignalen (L en H) kortsluiting zal optreden.

  Men sluit registeruitgangen dan ook via speciale bufferversterktrappen aan op een databus. De uitgangen van deze buffers kunnen met behulp van een z.g. ENABLE - commando in een HOOGOHMIGE TOESTAND worden gebracht; dan zijn deze uitgangen als het ware losgeschakeld van de databus.



Signaal OE commandeert of de informatie al of niet wordt AFGEGEVEN.

- Een UNI-DIRECTIONAL databus is een databus, waarover de data slechts in één richting kunnen worden verstuurd.

- Een BI-DIRECTIONAL databus is een databus, waarover de data óf in de ene, óf in de andere richting verstuurd kunnen worden. Daartoe is nodig dat uitgangen via buffers met hoogohmige toestand op de databus zijn aangesloten.

- Informatie kan worden opgeslagen in een register. Via een INGANGSPOORT kunnen data aan de registers worden toegevoerd. Ook via een UITGANGSPOORT kunnen de data van de registers worden afgenomen.

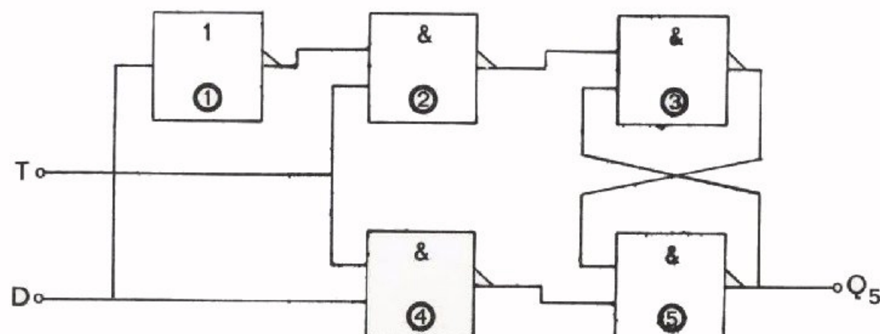
4

D E   C O N T R O L E B U S

In de vorige les heeft U geleerd op welke wijze een aantal registers parallel aangesloten kunnen worden op de databus. De registers kunnen dan, als ze geselecteerd zijn, informatie van de databus opnemen of weer op de databus zetten. Deze selectie gebeurt door de klokingang en door de enable ingang voor de "hoogohmige toestand" mogelijkheid. Omdat dit samengaan van registers met een hoogohmige toestand uitgang veel voorkomt, zijn er IC's ontwikkeld die bijvoorbeeld latch-flip-flops bevatten met een hoogohmige toestand uitgang. In het eerste deel van deze les zullen we een dergelijke flip-flop bekijken.

LATCH-FLIP-FLOP MET HOOGOHMIGE TOESTAND

Hieronder is een latch-flip-flop weergegeven, die is opgebouwd uit vier inverterende AND-poorten en één inverter.



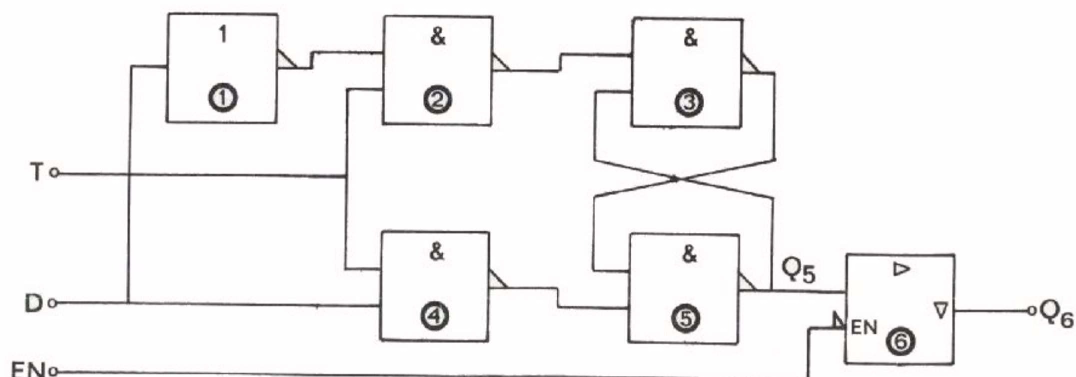
U herinnert zich: Q_5 is H als D = H en T = H

Q_5 is L als D = L en T = H

Dit is herhaald in onderstaande functietabel.

T	D	$Q_{5(n+1)}$	
L	L	Q_{5n}	Onthoud toestand
L	H	Q_{5n}	
H	L	L	Reset toestand
H	H	H	Set toestand

We gaan het circuit uitbreiden met een buffer met een hoogohmige toestand uitgang zoals hieronder. Ook bepalen we de functietabel.



	EN	T	D	$Q_{5(n+1)}$	$Q_{6(n+1)}$
1	L	L	L	Q_{5n}	Q_{5n}
2	L	L	H	Q_{5n}	Q_{5n}
3	L	H	L	L	L
4	L	H	H	H	H
5	H	L	L	Q_{5n}	Z
6	H	L	H	Q_{5n}	Z
7	H	H	L	L	Z
8	H	H	H	H	Z

Als de ingang EN hoog is, dan is de uitgang niet vrijgegeven en verkeert deze in de hoogohmige toestand (regels 5 tot en met 8 van bovenstaande functietabel).

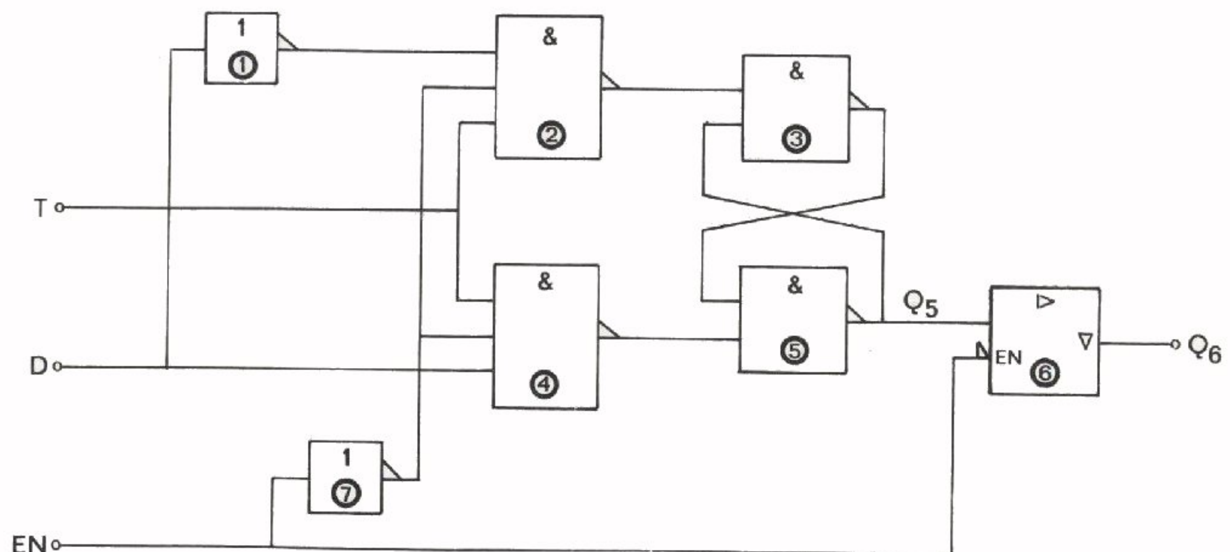
Als EN hoog is kan echter in de latch-flip-flop een informatie opgeslagen worden, mits T = H (regels 7 en 8). Op regel 5 en 6 wordt de informatie (L of H), die reeds opgeslagen was, onthouden maar niet beschikbaar gesteld aan uitgang Q_6. Dit laatste gebeurt op regels 1 en 2. Dus:

als EN laag is <u>en</u> T laag is.

De situaties op regels 3 en 4 zijn echter ongewenst als de latch-flip-flop aan de databus is aangesloten. Dus:

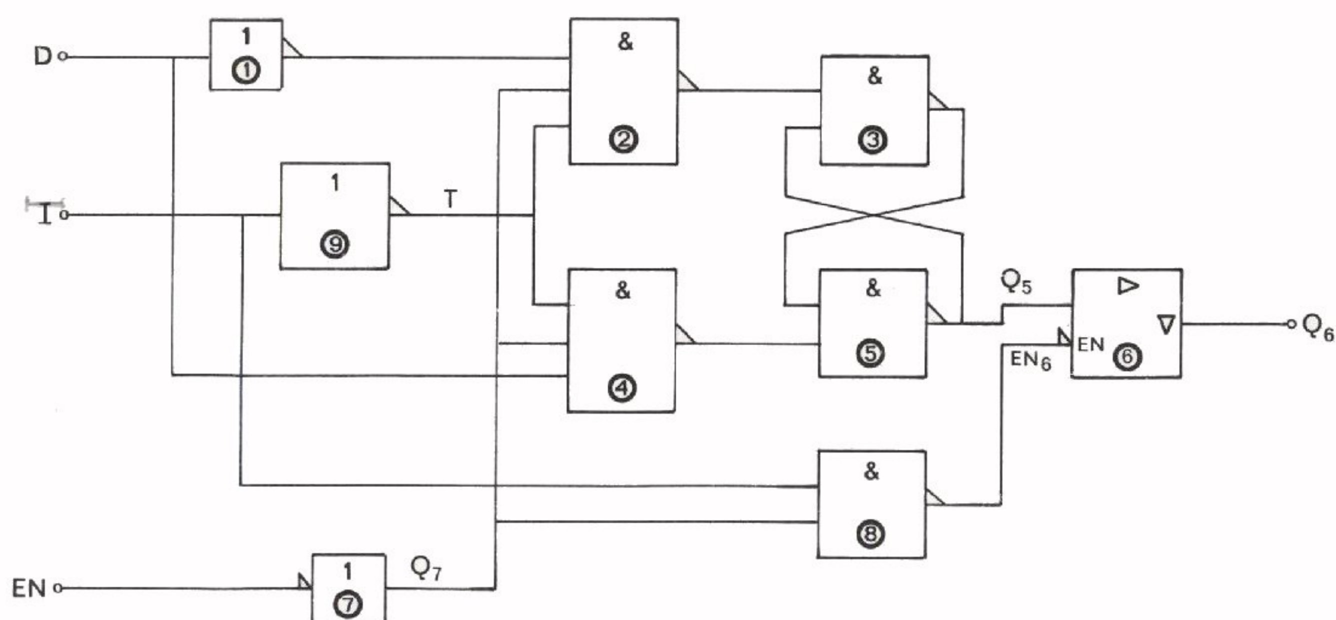
als EN laag is <u>en</u> T hoog is.

In deze gevallen, wordt ingang D met uitgang Q_6 kortgesloten via de databus. Dit komt omdat de EN-ingang uitsluitend de uitgang van de latch-flip-flop selecteert. We kunnen het circuit zodanig wijzigen dat de EN-ingang ook het opslaan van informatie in de latch-flip-flop beïnvloedt. Dit gaan we in twee fasen doen.



	EN	T	D	$Q_{5(n+1)}$	$Q_{6(n+1)}$
1	L	L	L	Q_{5n}	Q_{5n}
2	L	L	H	Q_{5n}	Q_{5n}
3	L	H	L	L	L
4	L	H	H	H	H
5	H	L	L	Q_{5n}	Z
6	H	L	H	Q_{5n}	Z
7	H	H	L	Q_{5n}	Z
8	H	H	H	Q_{5n}	Z

De enige verandering is dat als EN hoog is, de informatie niet meer kan worden opgeslagen en ook niet aan uitgang Q_6 worden doorgegeven. Als EN hoog wordt, verblijft de latch-flip-flop in de onthoudtoestand, onafhankelijk van de toestand van ingangen T en D (regels 5 tot 8).
Het nadeel van "zodra men in de latch-flip-flop een informatie opslaat, dan wordt deze informatie tegelijkertijd aan de uitgang afgegeven" wordt echter niet opgelost.
Dit nadeel is te vermijden door de schakeling nog enigszins uit te breiden zoals hieronder is gedaan.
De klokingang wordt ook aangesloten aan de EN-ingang van buffer ⑥ via de inverterende AND ⑧.



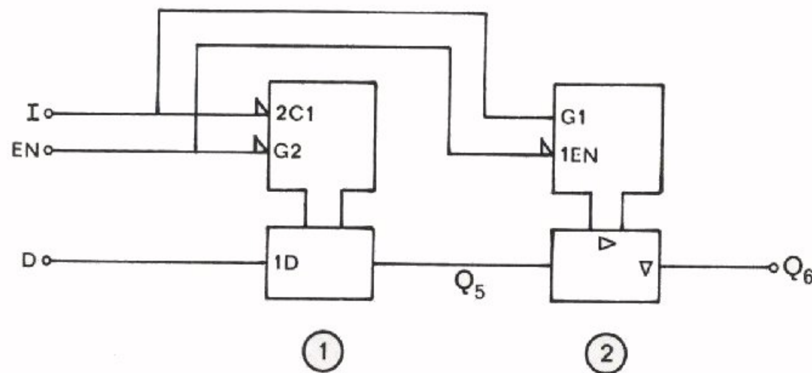
De werking kunnen we weer nagaan met behulp van de functietabel.

	EN	D	I	T	EN_6	$Q_{5(n+1)}$	$Q_{6(n+1)}$
1	L	L	L	H	H	L	Z
2	L	L	H	L	L	Q_{5n}	Q_{5n}
3	L	H	L	H	H	H	Z
4	L	H	H	L	L	Q_{5n}	Q_{5n}
5	H	L	L	H	H	Q_{5n}	Z
6	H	L	H	L	H	Q_{5n}	Z
7	H	H	L	H	H	Q_{5n}	Z
8	H	H	H	L	H	Q_{5n}	Z

Nu hebben we eindelijk ons doel bereikt. Als EN laag is en ook I laag is, kunnen we in de schakeling het niveau van ingang D opslaan. Dit niveau is echter niet beschikbaar aan uitgang Q_6 want deze verkeert in de hoogohmige toestand (regels 1 en 3). De vorige opgeslagen informatie is aan uitgang Q_6 beschikbaar als EN laag is en I hoog is, onafhankelijk van het niveau aan ingang D (regels 2 en 4). Voor de andere mogelijkheden (regels 5 t/m 8) blijft de flip-flop in de onthoudtoestand en de uitgang Q_6 in de hoogohmige toestand.

SYMBOLIEK

De schakeling die we net bestudeerd hebben kan, volgens de normen die in hoofdstuk 1 beschreven werden, veel eenvoudiger getekend worden. Bekijk zorgvuldig onderstaande tekening.



Laten we eerst het element ① bekijken. In het commandoblok ziet U verschillende notaties die we nu gaan uitleggen.

G2 U moet eerst weten dat de letter G gereserveerd is voor een AND-afhankelijkheid. Het cijfer 2 is een volgordenummer, willekeurig gekozen. Het betekent echter dat een andere ingang die ook het cijfer 2 bevat, beïnvloed is door deze controle-ingang. In ons geval is dit ingang 2C1.

2C1 Dit is de klokingang. Deze is uitsluitend actief als, intern in het blokje bekeken, G2 "1" is. Als dan ook I laag is, dus intern gelijk aan "1" (denk aan het polariteitsvlaggetje), zal Q_5 gelijk zijn aan het niveau van ingang D.

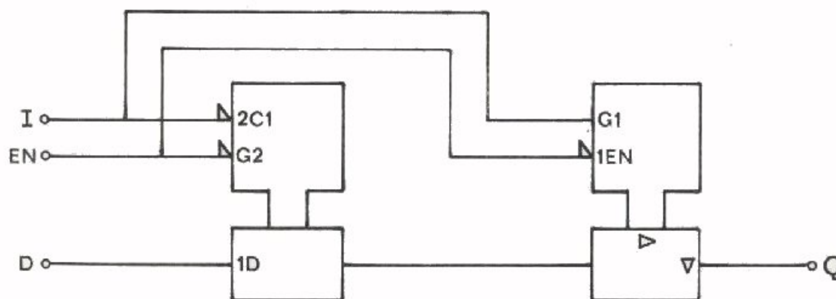
In de andere gevallen blijft de latch-flip-flop in de onthoudtoestand. Dus

$$Q_{5(n+1)} = Q_{5n}$$

Dit komt overeen met de vorige functietabel. Wanneer wordt nu $Q_6 = Q_5$? Met andere woorden, wanneer is de uitgang van element ② vrijgegeven? Dit kunnen we lezen in het commandoblok van element ②.
De EN-ingang (actief laag) zit in een AND-afhankelijke toestand met ingang I (actief hoog). Dus, als EN laag is en I hoog is, zal Q_6 gelijk zijn aan de toestand van Q_5. In alle andere toestanden zal de uitgang Q_6 van element ② in de hoogohmige toestand verkeren.

DE R/W-INGANG

Laten we het onderstaande circuit nog eens bekijken.



Het is een latch-flip-flop met een hoogohmige uitgang mogelijkheid. Wanneer ingang EN laag is, kan er informatie (1 bit) in de latch-flip-flop opgeslagen of aan de uitgang beschikbaar gesteld worden, afhankelijk van het niveau van ingang I. Als EN hoog is dan is de latch-flip-flop volkomen geïsoleerd.

Laten we aannemen dat EN laag is. Als I laag is kan informatie (1 bit) in de latch-flip-flop opgeslagen worden. Men zegt dat data in de flip-flop geschreven kan worden. Dus:

- De flip-flop is geselecteerd (EN is laag) en verkeert in de schrijftoestand (I is laag).

Ingang EN is nog steeds laag maar nu is I hoog. De reeds opgeslagen informatie is nu aan uitgang Q beschikbaar. Men zegt dat data uit de latch-flip-flop gelezen kan worden.

- De flip-flop is geselecteerd want EN is laag en verkeert in de leestoestand (I is hoog).

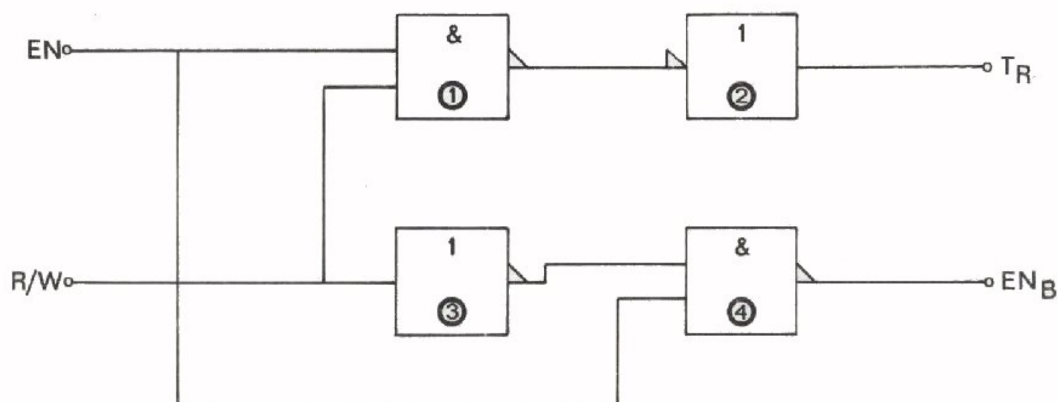
Dus, ingang I bepaalt, zodra de latch-flip-flop geselecteerd is, of er in de latch-flip-flop geschreven wordt (I is laag) of uit de latch-flip-flop gelezen wordt (I is hoog). Deze ingang I wordt R/W-ingang genoemd.

R staat voor "to read" (lezen).
W staat voor "to write" (schrijven).

De EN- en R/W-ingangen hebben een latch-flip-flop volledig onder controle. In een register, samengesteld uit een aantal latch-flip-flops, zijn de EN-ingangen met elkaar doorverbonden, evenals de R/W-ingangen.
Als op een databus meerdere registers zijn aangesloten, dan moeten we er wel voor zorgen, dat slechts één van deze registers op de bus kan schrijven, terwijl de anderen lezen wat er op de bus staat of er helemaal van los geschakeld zijn. De EN-en R/W-ingangen van elk register worden daarom door een centrale eenheid gestuurd. De EN-ingangen selecteren een bepaald register, de R/W-ingangen bepalen dat het geselecteerd register zal schrijven op de databus of lezen van de databus. Deze bundel van EN-en R/W-ingangen, de controlelijnen, vormen samen de CONTROLEBUS.

In het volgende experiment gaan we de theorie van de vorige pagina's toepassen op het circuit van pagina 3.18. We beschikken echter, in onze onderdelendoos, niet over een latch-flip-flop zoals net beschreven. We gaan de combinatie register-buffer van het vorige hoofdstuk besturen met een schakeling die hetzelfde effect heeft als op de vorige pagina's is beschreven. De uitbreiding zal stap voor stap plaats vinden. U herinnert zich, dat een 2-bits informatie van het ene register naar het andere kon schuiven door middel van de T- en EN-controle-ingangen. In feite was het T-signaal de klokingang van het 2-bits register en EN de enable ingang van de buffer. Met dit circuit was het echter ook mogelijk om tegelijkertijd te SCHRIJVEN (T = H) en te LEZEN (EN = L).
Dit gaan we proberen te vermijden met het circuit van de volgende pagina.

EEN CONTROLE CIRCUIT



Bovenstaand circuit is zodanig uitgevoerd met inverterende AND-poorten, dat men het later met de 7400-module kan opbouwen. We gaan dit analyseren met de volgende functietabellen.

EN	R/W	EN.R/W	Q_1	T_R		EN	R/W	Q_3	Q_3.EN	EN_B
L	L	L	H	L	1	L	L	H	L	H
L	H	L	H	L	2	L	H	L	L	H
H	L	L	H	L	3	H	L	H	H	L
H	H	H	L	H	4	H	H	L	L	H

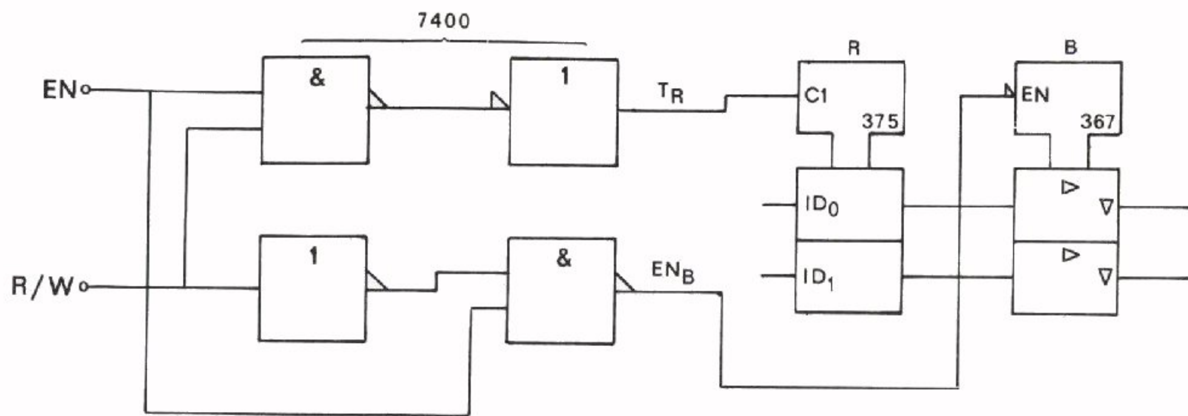
Uitgang T_R (tabel links) is L behalve als EN en R/W tegelijk hoog zijn.

Dus T_R = EN(H).R/W(H)

Uitgang EN_B (tabel rechts) is H behalve als EN is H en R/W is L.

Dus EN_B = EN(H).R/W(L)

We gaan bovenstaand circuit gebruiken met een combinatie register/buffer met hoogohmige toestand.



- Als EN laag is, is EN_B altijd H. Met andere woorden de uitgang van de buffer is in de hoogohmige toestand en er kan niet in de registercombinatie RB gelezen worden. Tegelijkertijd is T_R laag en er kan ook niet in R geschreven worden. Dus met EN laag is de combinatie RB volledig geïsoleerd.

- In regel 3 hebben we EN = H en R/W = L.
  In dit geval is T_R laag en kan er dus niet in het register geschreven worden. EN_B is echter L, dus kan er wel uit de combinatie RB gelezen worden.
  Dus: EN = H en R/W = L is de LEES toestand.

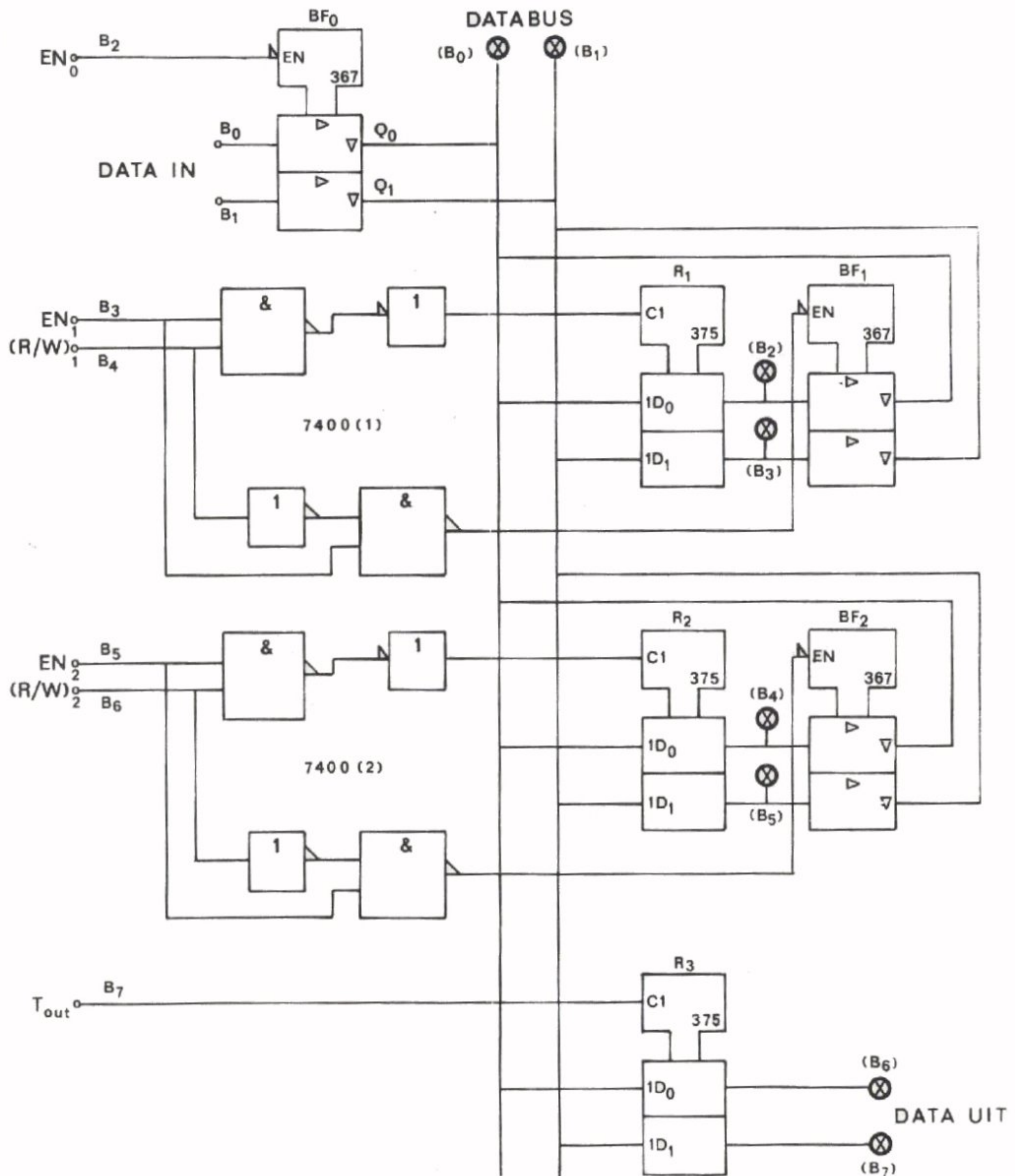
- In regel 4 is EN = H en R/W = H. EN_B = H en de uitgang van de combinatie RB is geïsoleerd. Maar nu is T_R = H en het register kan informatie opslaan. Er kan dus in het register geschreven worden.
  Dus: EN = H en R/W = H is de SCHRIJF toestand.

CONCLUSIE:

- De EN-ingang selecteert het register (combinatie van een 2-bits register en een buffer met hoogohmige toestand).

- De R/W-ingang bepaalt of er uitgelezen of er ingeschreven wordt.

In het volgende experiment gaan we het circuit van pagina 3.18 uitbreiden met het controle circuit, dat we op de vorige pagina's hebben bestudeerd. De uitbreiding vinden we in het circuit hieronder.



MODULE	AANSLUITINGEN											
8-BIT SELECTOR	B_3	B_4	B_5	B_6								
74367 (BF_0)	R	R	Ge	Ge								
74LS375 (R_1)	│	│	│	│	C_1							
74LS375 (R_2)	│	│	│	│	│		C_1					
74367 (BF_1)	│	│	│	│	O	OE_2	O					
74367 (BF_2)	│	│	│	│	│	O	│	OE_1				
74LS375 (R_3)	│	│	│	│	│	│	│	│				
8-LED INDICATOR	│	│	│	│	│	│	│	R				
7400 ①★	G_5 —R— G_4	G_6 —R— G_2	│	│	Q_4	Q_2	│	│	G_7 —R— Q_3	Q_1 —KB— G_3	G_1 —KS— P	G_8 —KB— P
7400 ②★			G_5 —R— G_4	G_6 —R— G_2			Q_4	Q_2	G_7 —R— Q_3	Q_1 —KB— G_3	G_1 —KS— P	G_8 —KB— P
DATA BUS												

KS = 2-polige kortsluitsteun

KB = kortsluitingsbeugel

★ 7400 ① is de module links op het schakelpaneel
7400 ② is de module rechts op het schakelpaneel

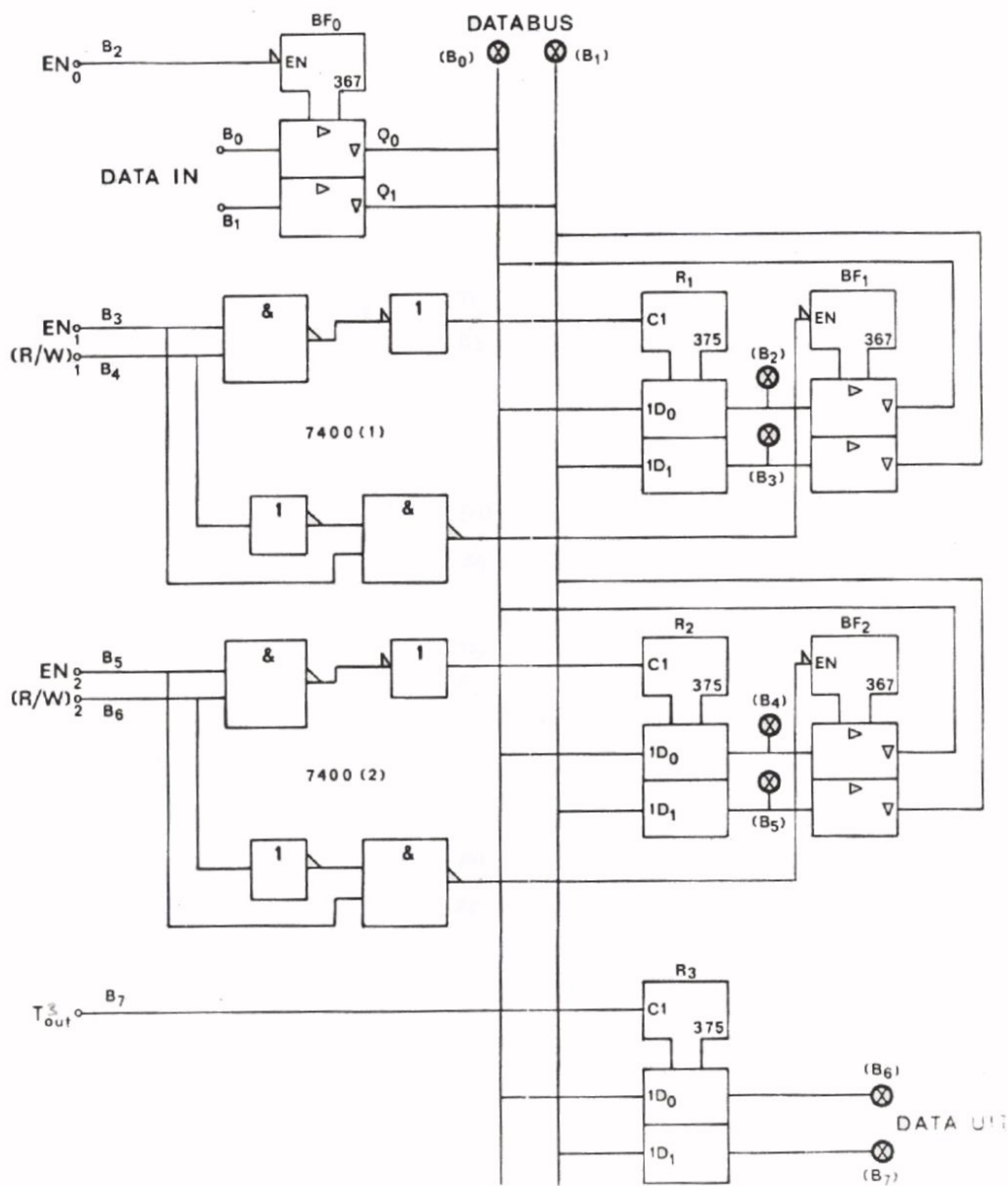
OPDRACHT 4.1: HET BESTUREN VAN REGISTERS (A)

- Plaats twee 7400 modulen op het schakelpaneel, tussen de 8-BIT selector en de 8-LED indicator.
- Verwijder de verbindingssnoeren aan de uitgangen B_3, B_4, B_5, B_6 van de 8-BIT selector. Wijzig daarna het circuit zoals op het bedradingsschema hiernaast is weergegeven. Denk er aan dat, voor een inverterende AND poort, de niet gebruikte ingangen aan de +5 V (P) moeten worden aangesloten.

- Op de volgende twee pagina's ziet U links een herhaling van het circuit van pagina 4.9 en rechts een tabel die U proefsgewijze moet invullen. Het doel van deze opdracht is om vanaf de "buitenwereld" via de ingangspoort weer een 2-bits informatie op de data bus te zetten en in een of ander register in te schrijven. U kunt ook deze 2-bits informatie naar eigen keuze verplaatsen.

- Alvorens een register te selecteren met de EN ingang moet U altijd eerst bepalen wat het register moet doen: lezen of schrijven. Immers: als men begint met selecteren, gaat de schakeling onmiddellijk lezen of schrijven, terwijl achteraf misschien andersom de bedoeling was. Eerst dus de R/W ingang coderen en daarna de EN ingang op H zetten. Probeer bijvoorbeeld de inhoud van R_1 op R_3 te zetten.

- Breek aan het einde van de opdracht de schakeling NIET af.


DATABUS
BF0
EN
367
(B0)
(B1)
EN0
B2
B0
Q0
DATA IN
Q1
B1
R1
BF1
EN1
B3
(R/W)1
B4
&
1
C1
375
EN
367
(B2)
1D0
7400 (1)
1D1
(B3)
R2
BF2
EN2
B5
(R/W)2
B6
(B4)
7400 (2)
(B5)
R3
T3out
B7
(B6)
DATA UIT
(B7)

Aansluitingen op 8-BIT selector en 8-LED indicator

CONTROLEN								UITGANGEN								
DATA		EN_0	EN_1	$(R/W)_1$	EN_2	$(R/W)_2$	T_{OUT}	DATABUS		R_1		R_2		OUT		
B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	(B_0)	(B_1)	(B_2)	(B_3)	(B_4)	(B_5)	(B_6)	(B_7)	
L	L	H	L	L	L	L	L	x	x	x	x	x	x	x	x	1
H	L	H	L	L	L	L	L	x	x	x	x	x	x	x	x	2
H	L	L	L	L	L	L	L			x	x	x	x	x	x	3
H	L	L	L	H	L	L	L			x	x	x	x	x	x	4
H	L	L	H	H	L	L	L					x	x	x	x	5
H	L	L	L	H	L	L	L					x	x	x	x	6
H	L	H	L	H	L	L	L					x	x	x	x	7
L	H	H	L	H	L	L	L					x	x	x	x	8
L	H	L	L	H	L	L	L					x	x	x	x	9
L	H	L	L	H	L	H	L					x	x	x	x	10
L	H	L	L	H	H	H	L							x	x	11
L	H	L	L	H	L	H	L							x	x	12
L	H	H	L	H	L	H	L							x	x	13
H	H	H	L	H	L	H	L							x	x	14
H	H	L	L	H	L	H	L							x	x	15
H	H	L	L	H	L	H	H									16
H	H	L	L	H	L	H	L									17
H	H	H	L	H	L	H	L									18

CONCLUSIES

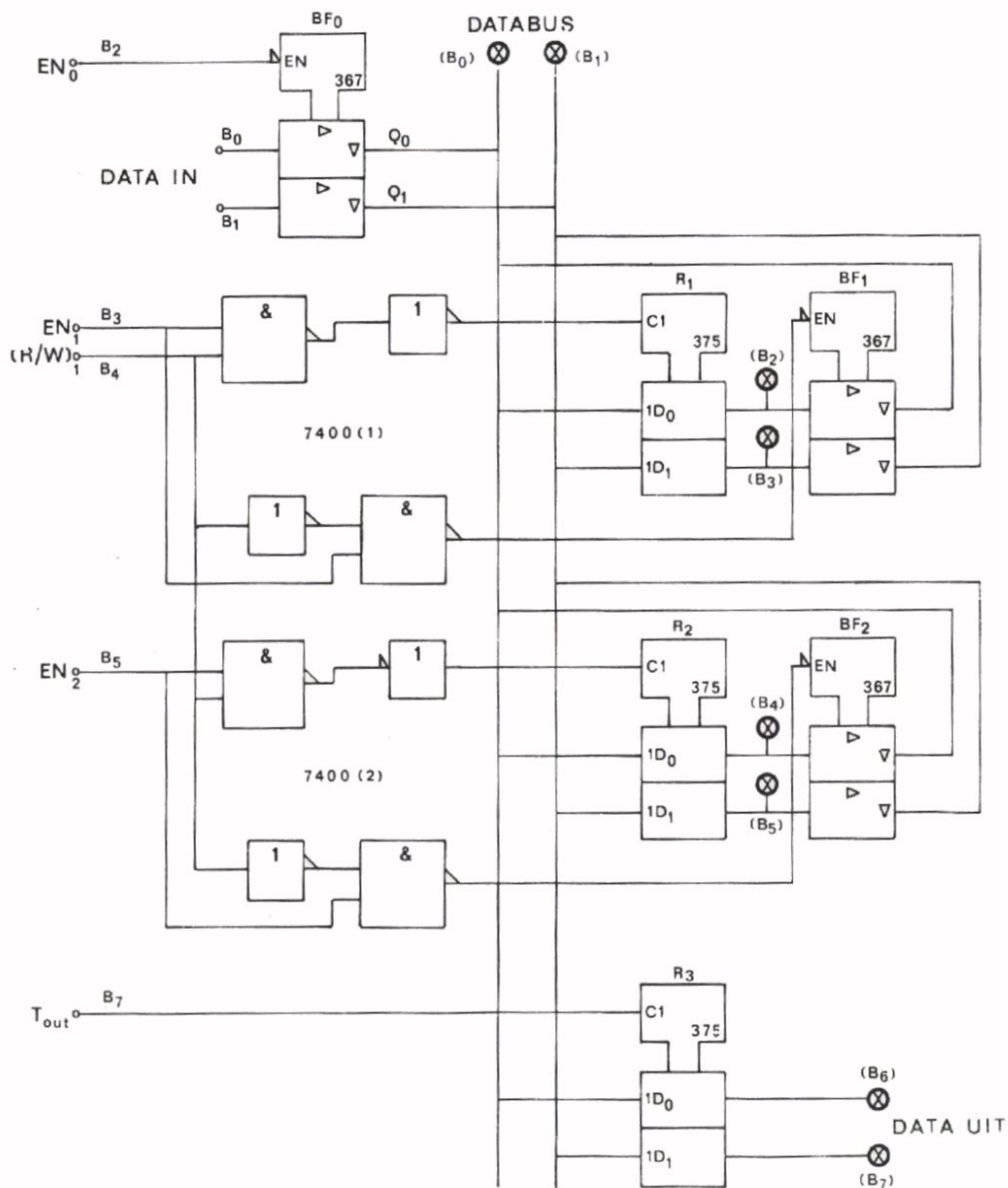
Zoals in opdracht 3.4 hebben we informatie (data) laten verschuiven via de databus.

De aansluitingen B_2 tot en met B_7 op de 8-BIT selector vormen de controlebus, die U zelf moet programmeren. Met andere woorden, U moet zelf opdrach geven aan iedere buffer of elk register wat deze moet doen, zoals:

- data van de "buitenwereld" op de databus zetten
- schrijven op de databus
- lezen van de databus
- informatie van de databus aan de "buitenwereld" beschikbaar stellen.

In de opdracht hadden we registers R_1 en R_2 onder controle met de EN ingangen. Het is niet mogelijk in het register te schrijven of te lezen als dit niet geselecteerd is. De enable ingang EN, bepaalt welk register gebruikt zal worden.

Ingangen EN_0, EN_1, EN_2, T_{OUT}. $(R/W)_1$ en $(R/W)_2$ vormen tezamen de CONTROLEBUS. De ingangspoort en de uitgangspoort hebben elk een controlelijn, met andere woorden, ze hebben elk een commando nodig. Registers R_1 en R_2 hebben elk twee commando's nodig: R/W en EN.
Deze controlebus kan vereenvoudigd worden door de R/W ingangen door te verbinden. Dit gaan we in het volgende experiment zien.


BF0
DATABUS
EN0
B2
EN
367
(B0)
(B1)
B0
Q0
DATA IN
Q1
B1
EN1
B3
(R/W)1
B4
&
1
R1
C1
375
BF1
EN
367
(B2)
1D0
1D1
(B3)
7400(1)
1
&
EN2
B5
&
1
R2
C1
375
BF2
EN
367
(B4)
1D0
1D1
(B5)
7400(2)
1
&
Tout
B7
R3
C1
375
(B6)
1D0
1D1
DATA UIT
(B7)

OPDRACHT 4.2: HET BESTUREN VAN REGISTERS (B)

- Het R/W-controlesignaal van R_2 wordt gestuurd door de aansluiting B_6 op de 8-BIT selector. Maak de aansluiting op B_6 los en steek de pen in B_4. In feite hebben we nu de R/W-controle-ingangen van R_1 en R_2 doorverbonden. Dit is op het circuit hiernaast weergegeven.

- Vul onderstaande tabel proefsgewijze in.

Aansluitingen op 8-BIT selector en 8-LED indicator

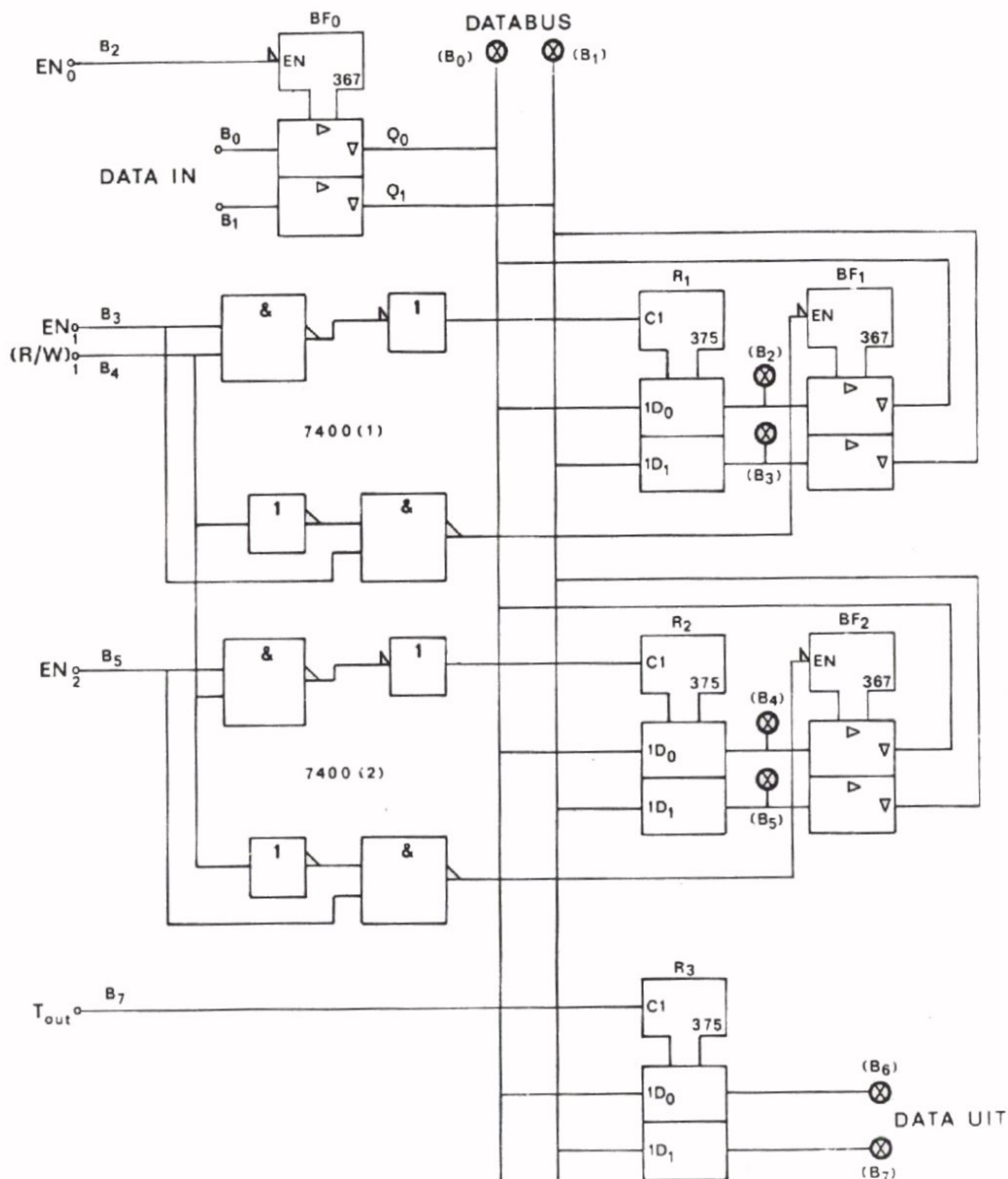
		CONTROLEN					UITGANGEN							
DATA		EN	EN_1	R/W	EN_2	T_{out}	DATA BUS		R_1		R_2		OUT	
B_0	B_1	B_2	B_3	B_4	B_5	B_7	(B_0)	(B_1)	(B_2)	(B_3)	(B_4)	(B_5)	(B_6)	(B_7)
L	L	H	L	L	L	L	x	x	x	x	x	x	x	x
H	L	H	L	L	L	L	x	x	x	x	x	x	x	x
H	L	L	L	L	L	L			x	x	x	x	x	x
H	L	L	L	H	L	L			x	x	x	x	x	x
H	L	L	H	H	L	L					x	x	x	x
H	L	H	H	H	L	L					x	x	x	x
H	L	H	L	H	L	L					x	x	x	x
L	H	H	L	H	L	L					x	x	x	x
L	H	L	L	H	L	L					x	x	x	x
L	H	L	L	H	L	L					x	x	x	x
L	H	L	L	H	H	L							x	x
L	H	H	L	H	H	L							x	x
L	H	H	L	H	L	L							x	x
H	H	H	L	H	L	L							x	x
H	H	L	L	H	L	L							x	x
H	H	L	L	H	L	H								
H	H	H	L	H	L	H								
H	H	H	L	H	L	L								

- Breek de schakeling af.

Vergelijk nu de tabel die U zonet hebt ingevuld met de tabel van opdracht 4.1
Ze moeten gelijk zijn, ofschoon er maar één R/W-commando nodig was voor de registers R_1 en R_2.

Dit is vrij gemakkelijk te begrijpen:
Dank zij het controle-circuit dat we hebben ingevoerd, is een bepaald register geselecteerd met behulp van zijn EN-ingang. Het feit dat een register op "schrijven" of "lezen" staat, heeft geen betekenis zolang het niet geselecteerd is door de EN-controle-ingang.

Daarom kunnen we nu alle R/W-controle-ingangen doorverbinden.



BF0
DATABUS
EN0
B2
EN
367
(B0)
(B1)
B0
Q0
DATA IN
Q1
B1
R1
BF1
EN1
B3
&
1
C1
375
EN
367
(R/W)1
B4
(B2)
1D0
7400(1)
1D1
(B3)
1
&
R2
BF2
EN2
B5
&
1
C1
375
EN
367
(B4)
1D0
7400(2)
1D1
(B5)
1
&
R3
Tout
B7
C1
375
(B6)
1D0
DATA UIT
1D1
(B7)

SAMENVATTING

- Het op de databus zetten van binaire informatie gebeurt via buffer BF_0, die fungeert als een ingangspoort. Deze ingangspoort is een onderdeel, waarmede we informatie van buiten het BUS-SYSTEEM in het bus-systeem kunnen schrijven. Op een commando aan de EN-ingang van BF_0, wordt de informatie op de databus geschreven.

- Via register R_3 wordt de informatie, die zich binnen het bus-systeem bevindt, bereikbaar gemaakt voor circuits buiten het bus-systeem. Register R_3 fungeert als uitgangspoort. Een uitgangspoort is een onderdeel via welke de informatie buiten het bus-systeem beschikbaar kan komen. Op een commando aan de C_1-ingang van R_3, wordt de informatie op de uitgangen van R_3 geschreven.

- In het circuit hiernaast wordt het bus-systeem gevormd door de databus en twee registers met hoogohmige toestand uitgang (R_1/BF_1 en R_2/BF_2). Ieder register is geselecteerd in bijvoorbeeld een EN-ingang. We hebben in experiment 4.2 gezien, dat de R/W-ingangen van registers doorverbonden kunnen worden. Dus het transport van informatie in het bus-systeem wordt bepaald door een EN-controlesignaal per register en één R/W-controlesignaal voor registers samen.

- De ingangen EN_0, EN_1, EN_2, R/W en T_{out} vormen tezamen de CONTROLEBUS. De 8-BIT selector, die U zelf programmeert, kan in het circuit hiernaast beschouwd worden als de CENTRALE CONTROLE UNIT.

- In principe kan het aantal op de databus aangesloten registers naar believen worden uitgebreid. Het enige probleem zullen de controlelijnen (EN) vormen, want met het toenemende aantal registers neemt ook het aantal EN-lijnen toe. Dit probleem wordt in de praktijk opgelost met behulp van DECODERS.

5

G E H E U G E N S   E N   A D R E S B U S

INLEIDING

In de vorige les is uitgelegd dat registers die aan de uitgang voorzien zijn van een 3 state buffer, zowel met hun ingang als met hun uitgang op de databus kunnen worden aangesloten.

Elk register is te selecteren met behulp van een EN-ingang. Verder zijn alle R/W-ingangen doorverbonden, zodat men met één R/W-signaal kan volstaan, om óf in het geselecteerde register te schrijven, óf uit het geselecteerde register te lezen.

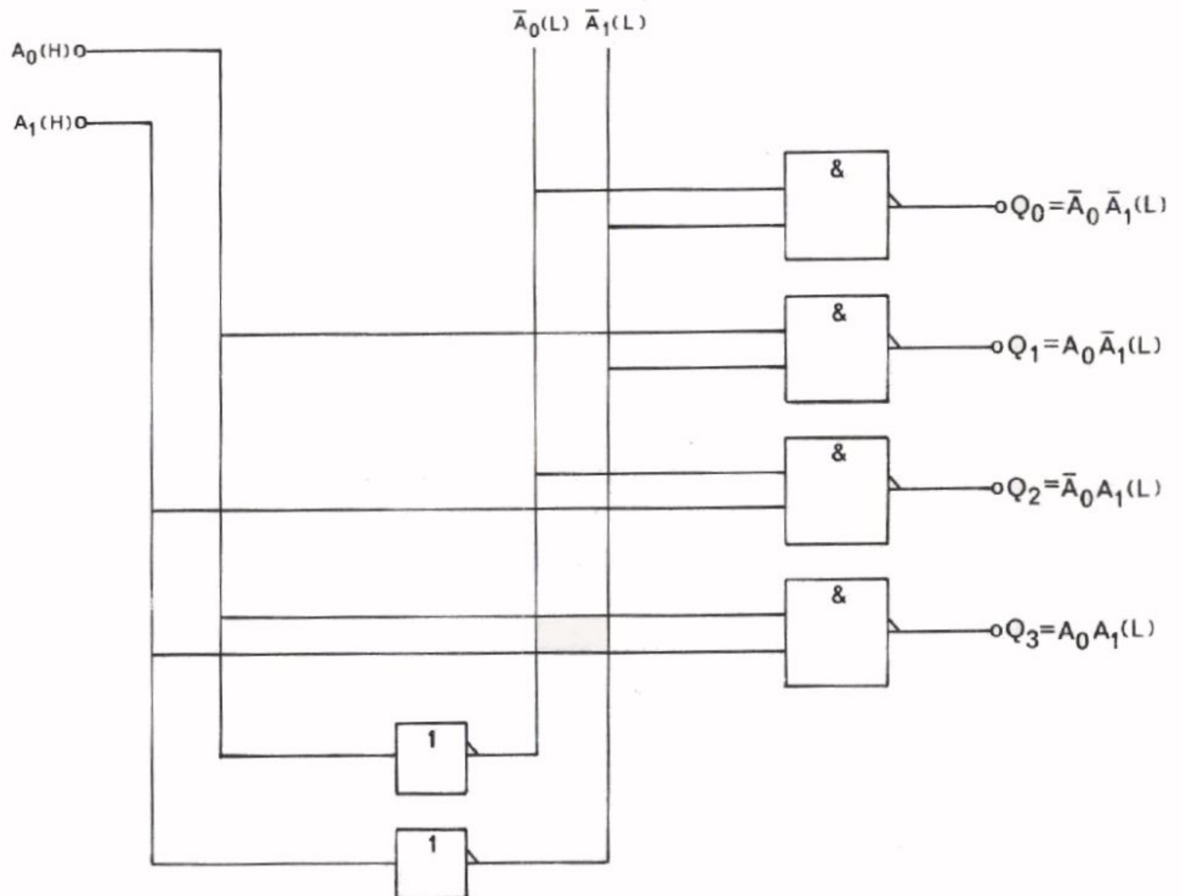
Heeft men een uit vele duizenden registers samengesteld geheugen, dan is het onpraktisch om deze registers m.b.v. even zovele duizenden EN-lijnen te selecteren. Door het toepassen van DECODERS kan men het aantal controlelijnen drastisch beperken. Dit wordt in het begin van deze les behandeld.

Daarna komen we op het "totaal van decoders en registers", het GEHEUGEN. Eerst bespreken we de RAM, het geheugen waarin geschreven en waaruit gelezen kan worden. Vervolgens de ROM, het geheugen waaruit men alleen maar kan lezen. Tenslotte de PROM, waarin eenmalig een vast programma geschreven kan worden.

Valt de voedingsspanning van de geheugens weg, dan gaat de in RAM's opgeslagen informatie verloren. De in ROM's en PROM's opgeslagen informatie blijft echter behouden.

DECODEERSCHAKELINGEN

In het D-traject werden in les D22 uitvoerig decodeerschakelingen behandeld. We zullen ons met onderstaande schakeling tot een korte herhaling beperken.



De functietabel van deze schakeling is als volgt:

A_1	A_0	Q_0	Q_1	Q_2	Q_3
L	L	L	H	H	H
L	H	H	L	H	H
H	L	H	H	L	H
H	H	H	H	H	L

Ga dit na!

Van de tabel kunnen we afleiden dat, voor elke binaire combinatie van A_0 en A_1, één van de uitgangen laag wordt, terwijl de overigen hoog zijn.

Een dergelijke schakeling wordt een decoder (code-omzetter) genoemd. Het vorige circuit is een binair naar 1-uit-4-decoder. Dit betekent:
de schakeling zet elk van de mogelijke binaire ingangscombinaties (in dit geval $2^2 = 4$ combinaties, namelijk LL, LH, HL en HH) om in een signaal (in dit geval L) op één van de vier uitgangen.

Decoders (of code-omzetter) worden zeer vaak toegepast en daarom bestaan ze ook in geïntegreerde vorm. U beschikt in de onderdelendoos over een decoder, de module 74138.

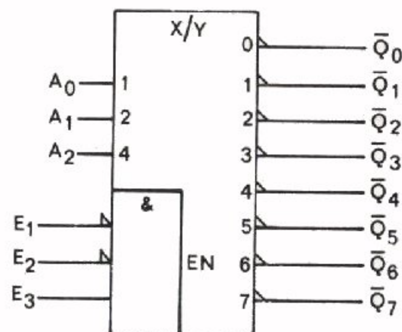
OPMERKING

Er bestaan ook andere decoders, zoals b.v.:

- BCD naar 1-uit-10 decoders,
- BCD naar 7-segment decoders,
- Excess 3 naar 1-uit-10 decoders.

DECODER 74138

De 74138 is een binair naar 1-uit-8-decoder. Dit betekent dat dit IC elk van de $2^3 = 8$ mogelijke binaire combinaties van de 3 ingangssignalen A_0, A_1 en A_2 om kan zetten in een uitgangssignaal op één van de 8 uitgangen.



A_2	A_1	A_0	$\overline{Q_0}$	$\overline{Q_1}$	$\overline{Q_2}$	$\overline{Q_3}$	$\overline{Q_4}$	$\overline{Q_5}$	$\overline{Q_6}$	$\overline{Q_7}$
L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	H	H	H	H	H	H
L	H	L	H	H	L	H	H	H	H	H
L	H	H	H	H	H	L	H	H	H	H
H	L	L	H	H	H	H	L	H	H	H
H	L	H	H	H	H	H	H	L	H	H
H	H	L	H	H	H	H	H	H	L	H
H	H	H	H	H	H	H	H	H	H	L

Hierboven ziet U het symbool van deze decoder en de functietabel.
Op het symbool worden alle uitgangen aangeduid met een streepje op de Q, want de uitgangen zijn actief als ze laag zijn.

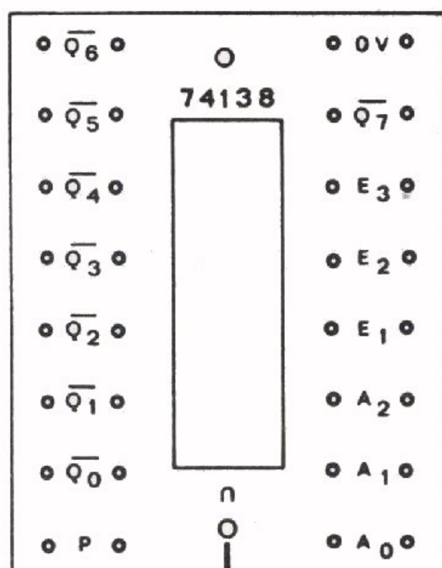
Op het symbool van de 74138 ziet U nog drie ingangen: E_1, E_2 en E_3.
Dit zijn drie output-enable ingangen. Om de uitgangen actief te kunnen maken moet aan de volgende eisen worden voldaan:

$E_1 = L$

$E_2 = L$

$E_3 = H$

MODULE 74138



Hiernaast ziet U een afbeelding van de bovenkant van de module, waarin het IC 74138 is gemonteerd. Evenals in de vorige modulen dienen de bodempennen voor de bevestiging op het schakelpaneel en voor het aansluiten van de voedingsspanning.

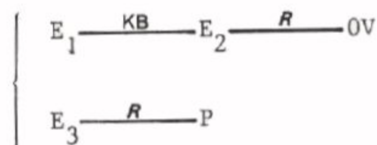
In het volgende experiment gaan we deze module testen.

OPDRACHT 5.1 DECODER-MODULE 74138

- Prik de 8-bit selector op het schakelpaneel. Prik de module 74138 rechts van de 8-bit selector en daarna de 8-LED indicator onder de module 74138.
- Het bedradingsschema is als volgt:

MODULE	AANSLUITINGEN										
8-BIT SELECTOR	B_0	B_1	B_2								
	Ge	Ge	Ge								
74138	A_0	A_1	A_2	$\overline{Q}_0$	$\overline{Q}_1$	$\overline{Q}_2$	$\overline{Q}_3$	$\overline{Q}_4$	$\overline{Q}_5$	$\overline{Q}_6$	$\overline{Q}_7$
				R	R	O	O	Ge	Ge	Ge	B
8-LED INDICATOR				B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7

Verder op de module 74138:



- Vul de volgende tabel proefsgewijze in:

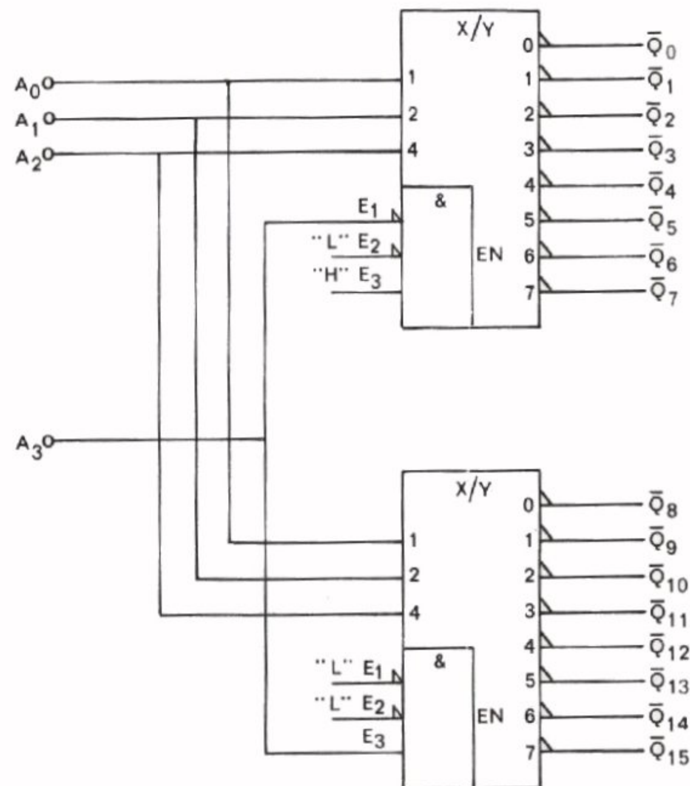
A_2	A_1	A_0	$\overline{Q}_0$	$\overline{Q}_1$	$\overline{Q}_2$	$\overline{Q}_3$	$\overline{Q}_4$	$\overline{Q}_5$	$\overline{Q}_6$	$\overline{Q}_7$
L	L	L								
L	L	H								
L	H	L								
L	H	H								
H	L	L								
H	L	H								
H	H	L								
H	H	H								

- Vergelijk deze tabel met die van pagina 5.4.
- Bovenstaande tabel is uitsluitend te realiseren wanneer E_1 = L, E_2 = L en E_3 = H. Ga dit na!
- Breek de schakeling af.

CONCLUSIE

Het IC 74138 is een binair naar 1-uit-8-decoder. De Q uitgangen kunnen om de beurt laag gezet worden met behulp van de A-ingangen.
Het is niet mogelijk twee uitgangen tegelijkertijd laag te zetten. Deze uitgangen worden gebruikt om bijvoorbeeld de "output enable" van registers te activeren. Dit gaan we in een volgende opdracht ervaren.

De drie enable ingangen van de decoder 74138 maken het mogelijk twee of meer decoders parallel te schakelen om bijvoorbeeld een binair naar 1-uit-16 of een binair naar 1-uit-24-decoder samen te stellen.



A_3	A_2	A_1	A_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}	Q_{14}	Q_{15}
L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

BINAIR NAAR 1-UIT-16-DECODER

Hiernaast ziet U een mogelijkheid om een binair naar 1-uit-16-decoder op te bouwen met twee parallel geschakelde decoders 74138. Als A_3 = L, wordt de bovenste decoder geselecteerd via de enable-ingang E_1, terwijl de uitgangen $\overline{Q}_8$ t/m $\overline{Q}_{15}$ in de H toestand verkeren want E_3 is laag. Het omgekeerde gebeurt waneer A_3 = H. De bovenste decoder zet dan zijn uitgangen hoog (E_1 = H) en de onderste decoder wordt geselecteerd (E_3 = H). De niet gebruikte enable-ingangen zijn constant in de actieve staat gezet: L voor E_1 en E_2, H voor E_3.

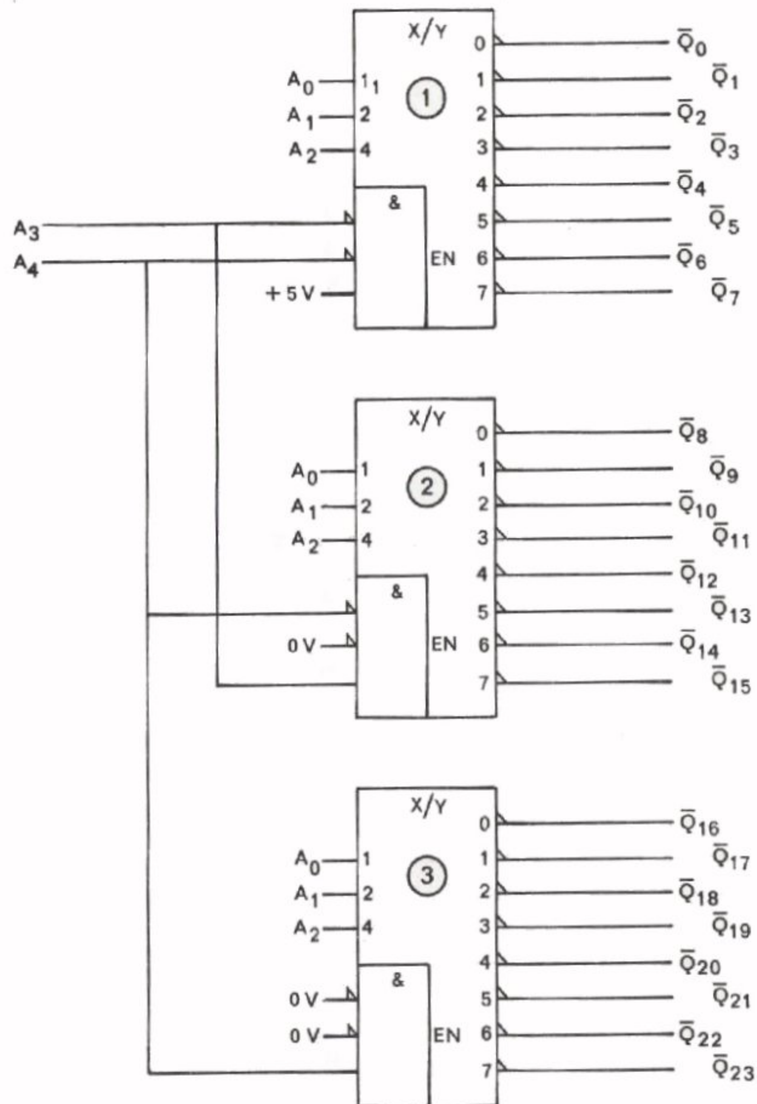
Hiernaast ziet U ook de functietabel van de binair naar 1-uit-16-decoder.

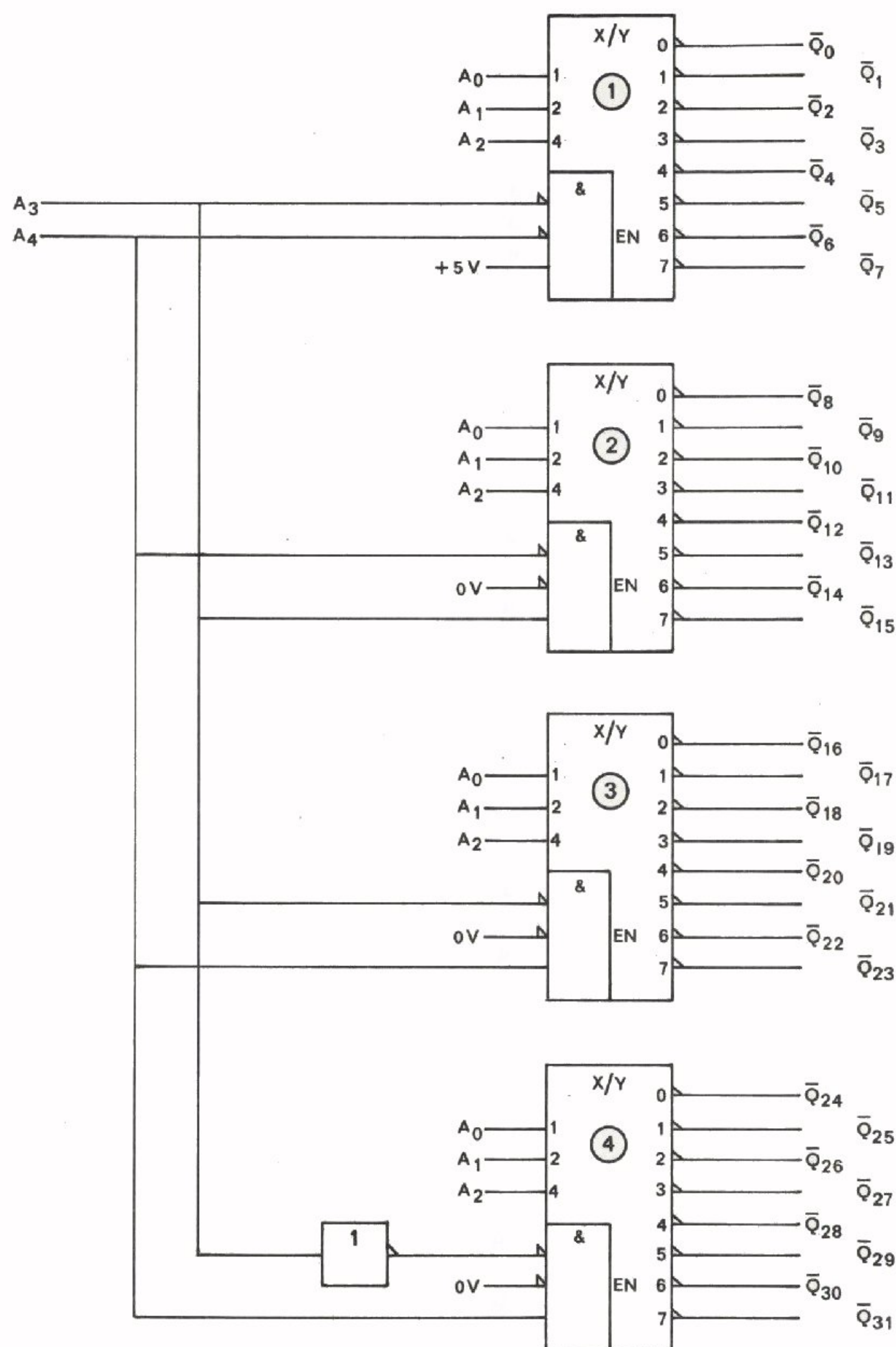
BINAIR NAAR 1-UIT-24-DECODER

Op de volgende pagina is een manier aangegeven om een binair naar 1-uit-24-decoder op te bouwen met drie decoders 74138. Ga dit na!

BINAIR NAAR 1-UIT-32-DECODER

Dankzij de drie enable-ingangen van de decoder 74138 is het ook mogelijk een binair naar 1-uit-32-decoder op te bouwen met vier decoders 74138. Hiervoor is echter een inverter extra nodig. Bekijk een mogelijke realisatie hiervan die op pagina 5.11 staat.


X/Y
1
2
3
A0
A1
A2
A3
A4
&
EN
+5 V
0 V
Q̄0
Q̄1
Q̄2
Q̄3
Q̄4
Q̄5
Q̄6
Q̄7
Q̄8
Q̄9
Q̄10
Q̄11
Q̄12
Q̄13
Q̄14
Q̄15
Q̄16
Q̄17
Q̄18
Q̄19
Q̄20
Q̄21
Q̄22
Q̄23


X/Y
A0
A1
A2
1
2
4
&
EN
0
1
2
3
4
5
6
7
A3
A4
+5 V
0 V
Q̄0
Q̄1
Q̄2
Q̄3
Q̄4
Q̄5
Q̄6
Q̄7
Q̄8
Q̄9
Q̄10
Q̄11
Q̄12
Q̄13
Q̄14
Q̄15
Q̄16
Q̄17
Q̄18
Q̄19
Q̄20
Q̄21
Q̄22
Q̄23
Q̄24
Q̄25
Q̄26
Q̄27
Q̄28
Q̄29
Q̄30
Q̄31
3

DATABUS

DB_0 DB_1

(B_0) (B_1)

BF_0

EN_0 B_2 EN 367

B_0 Q_0

DATA IN

B_1 Q_1

R/W B_5

1 & 7400 (1)

R_1 BF_1

C1 375 EN 367

(B_2) (B_3)

$1D_0$

$1D_1$

1 7402

& 1

X/Y 138

A_0 B_6 1

A_1 B_7 2

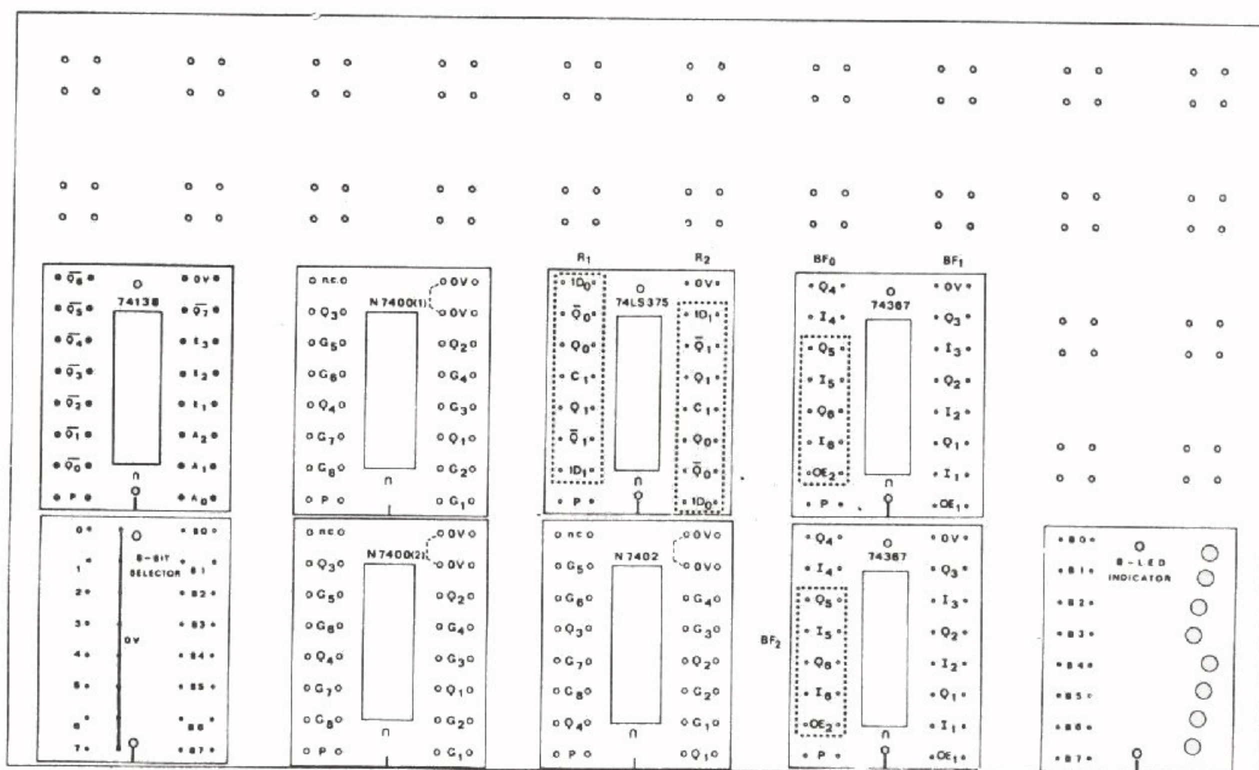
4

EN E_1 &

B_4 E_2 EN

E_3

0 1 2 3 4 5 6 7



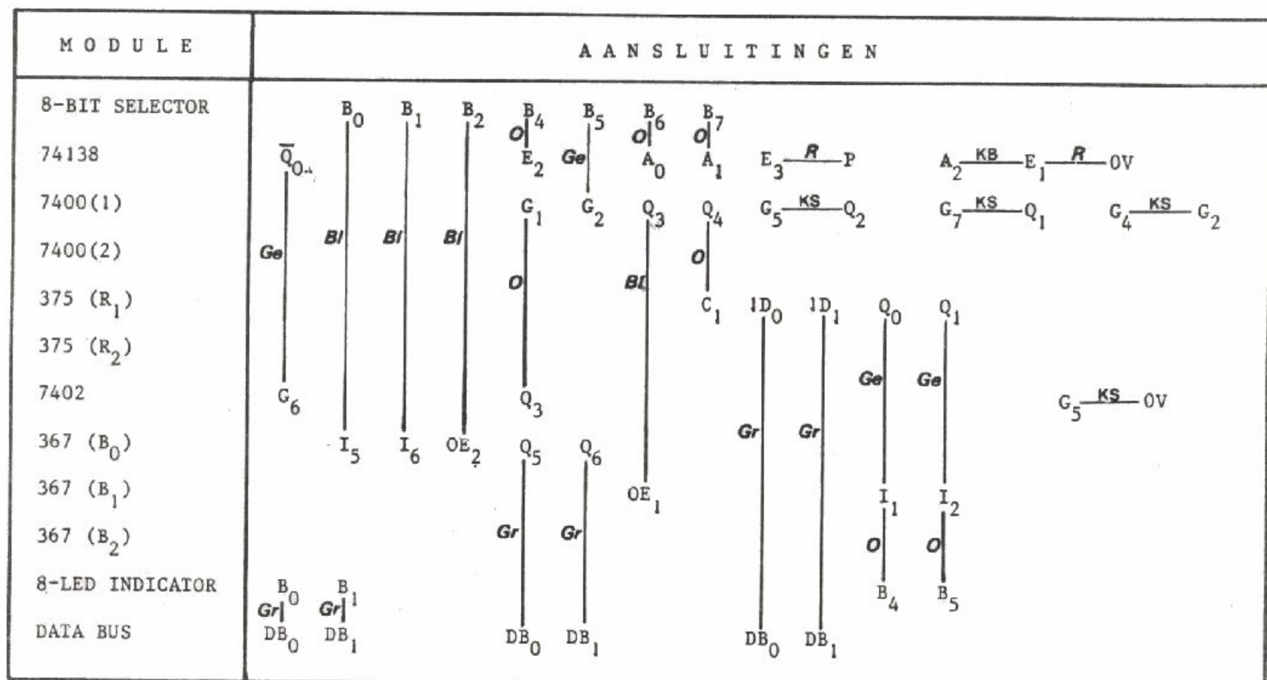
BESTUREN VAN REGISTERS

In de volgende opdracht gaan we een aantal registers opbouwen en ze selecteren via een decoder. In principe gaan we hetzelfde circuit gebruiken als dat van opdracht 4.2. We zullen echter de "uitgangspoort" niet meer aansluiten. De opdracht zal in drie fasen opgebouwd worden om een beter inzicht in het hele circuit te krijgen.

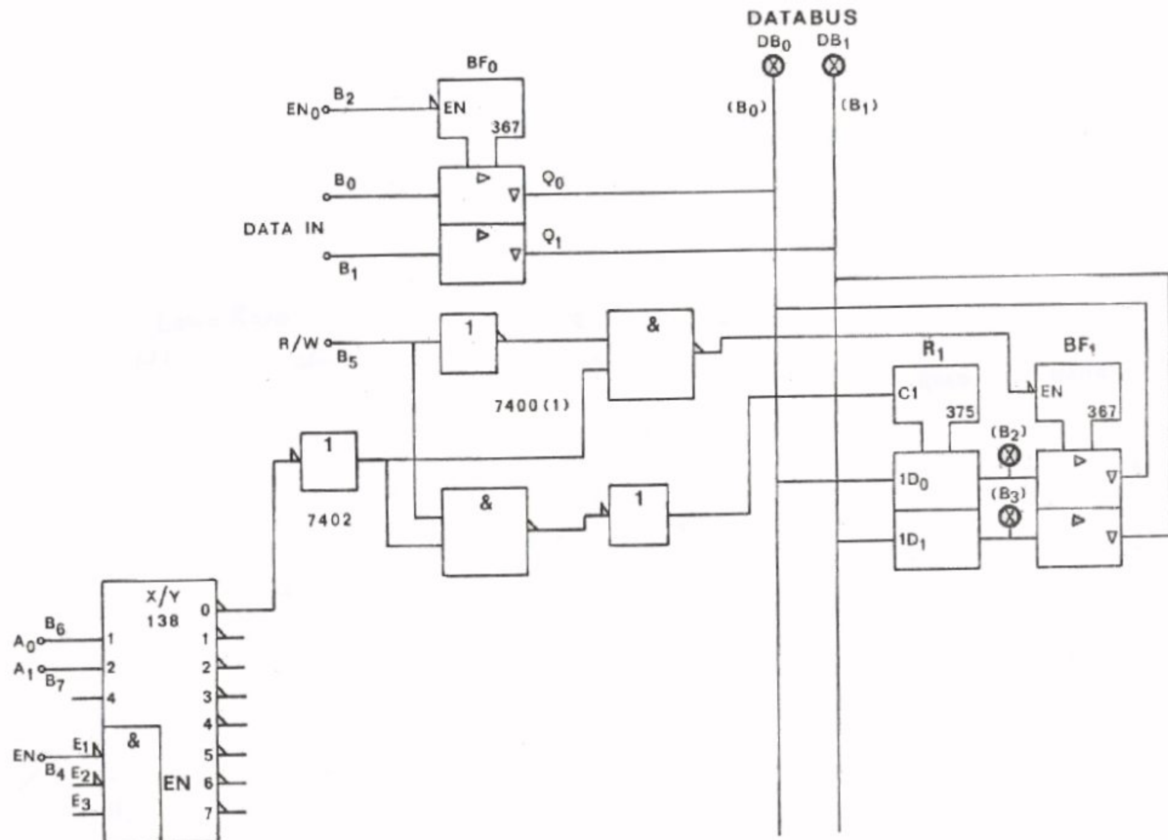
Hiernaast ziet U de locatie van de diverse modulen op het schakelpaneel en het circuit dat we in de eerste fase zullen bekijken.

OPDRACHT 5.2 REGISTERS EN DECODER (FASE 1)

- Het circuit bestaat uit een ingangspoort en een 2-bits register met controlecircuit. Dit register wordt geselecteerd met de decoder 74138. Er moet echter een inverter geplaatst worden tussen de decoder en het controlecircuit. U herinnert zich nog wel dat in opdracht 4.2 het register was geselecteerd door EN = H en dat de uitgangen van de decoder laag actief zijn.
- Het bedradingsschema is als volgt:



- Denk eraan, dat ongebruikte ingangen van de 7400 module met + 5 V doorverbonden moeten worden.
- Vul de volgende tabel proefsgewijze in. Breek daarna de schakeling nog NIET af.



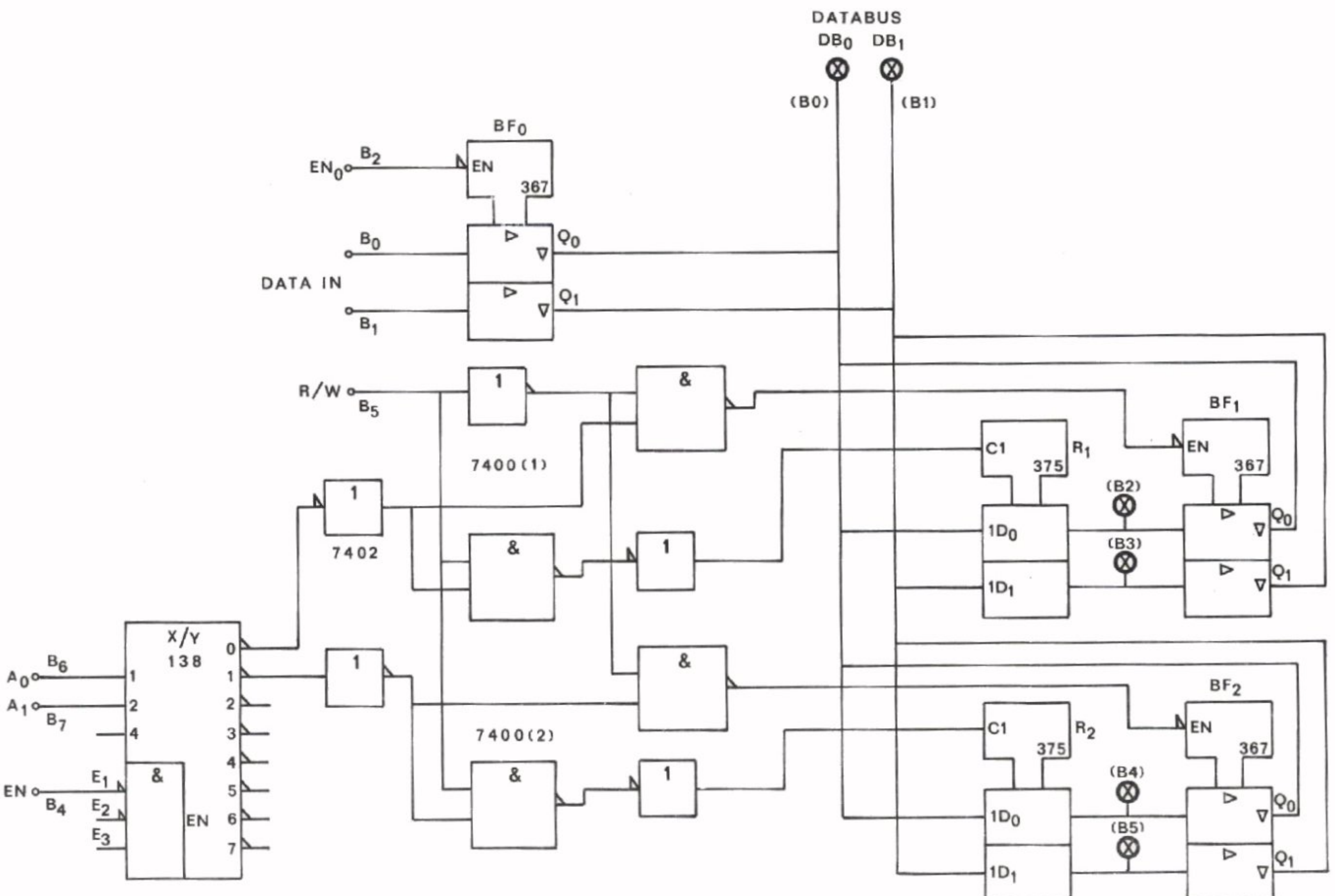
DATA		CONTROLEN					UITGANGEN				
		EN_0	R/W	EN	A_0	A_1	DATA BUS		R_1		
B_0	B_1	B_2	B_5	B_4	B_6	B_7	(B_0)	(B_1)	(B_2)	(B_3)	
L	L	H	L	H	L	L	x	x	x	x	1
H	L	H	L	H	L	L	x	x	x	x	2
H	L	L	L	H	L	L			x	x	3
H	L	L	H	H	L	L			x	x	4
H	L	L	H	L	L	L					5
H	L	L	H	H	L	L					6
H	L	H	H	H	L	L					7
L	L	H	H	H	L	L					8
L	L	H	L	H	L	L					9
L	L	H	L	L	L	L					10
L	L	H	L	H	H	L					11
L	H	H	L	H	H	L					12
L	H	L	L	H	H	L					13
L	H	L	H	H	H	L					14
L	H	L	H	L	H	L					15

ANALYSE VAN DE RESULTATEN

- In regel 1 zorgen we dat de buffer BF_0 en de decoder niet geselecteerd zijn. Hun uitgangen verkeren in de hoogohmige toestand.
- In regel 2 wordt een 2-bits data aan de ingang van buffer BF_0 aangesloten.
- In regel 3 wordt de data op de databus gezet.
- In regel 4 zorgen we dat het register in de "lees" toestand is.
- In regel 5 is de data van de databus in register R_1 ingeschreven.
- In regel 6 t/m 9 gaan we stapsgewijze terug naar de begintoestand (regel 1). U ziet dat in regel 9 het register R_1 nu in de "schrijf" toestand is geplaatst.
- In regel 10 wordt de data van R_1 op de databus gezet.
- In regel 11 hebben we de stand van ingang A_0 van de decoder gewijzigd. In regels 12 t/m 15 merken we dat het nu niet meer mogelijk is de data die op de databus is, in register R_1 op te slaan. Met andere woorden, register R_1 wordt niet meer geselecteerd.

CONCLUSIE

Register R_1 wordt door de decoder geselecteerd als A_0 en A_1 beide laag zijn.


DATABUS
DB_0 DB_1
(B0) (B1)
BF_0
EN_0 B_2
EN
367
B_0
Q_0
DATA IN
B_1
Q_1
R/W B_5
1
&
7400(1)
7402
7400(2)
X/Y
138
A_0 B_6
A_1 B_7
EN B_4
E_1 E_2 E_3
EN
0 1 2 3 4 5 6 7
C1
375
R_1
$1D_0$ $1D_1$
(B2) (B3)
BF_1
EN
367
Q_0 Q_1
C1
375
R_2
$1D_0$ $1D_1$
(B4) (B5)
BF_2
EN
367
Q_0 Q_1

OPDRACHT 5.2 REGISTERS EN DECODERS (FASE 2)

- We gaan nu het circuit uitbreiden met een tweede register, zoals hiernaast aangegeven.
- Het bedradingsschema voor de uitbreiding is als volgt:

M O D U L E	A A N S L U I T I N G E N
8-BIT SELECTOR	B_5
74138	$\overline{Q}1$
7400(1)	O
7400(2)	G_1 Q_2 Q_4 G_2 G_4 —R— P G_2 —KS— G_8 G_7 —KS— Q_1 —KB— G_3
375 (R_1)	O
375 (R_2)	C_1 KS $1D_0$ $1D_1$ Q_0 Q_1
7402	G8 Ge Q_4 G_7 —KS— G_5
367 (B_0)	Gr Gr R O
367 (B_1)	
367 (B_2)	OE_2 I_5 I_6
8-LED INDICATOR	O B_6 O B_7
DATA BUS	DB_0 DB_1

- Op dezelfde manier als in fase 1 kunt U data inschrijven in register R_2 of uitlezen van R_2. Hiervoor moet U wel uitgang $\overline{Q}_1$ van de decoder hoog zetten. Dit gebeurt met A_0 = H en A_1 = L.

- Breek de schakeling nog niet af.

DATABUS
DB_0 DB_1
(B_0) (B_1)
BF_0
EN_0 B_2 EN 367
B_0 Q_0
DATA IN
B_1 Q_1
R/W B_5
1 &
7400 (1)
7402
X/Y 138
A_0 B_6 1
A_1 B_7 2
4
EN B_4 E_1 E_2 E_3 & EN
0 1 2 3 4 5 6 7
7400 (2)
7400 (3)
7400 (4)
BF_1 BF_2 BF_3 BF_4
R_1 R_2 R_3 R_4
C1 375 $1D_0$ $1D_1$
EN 367 Q_0 Q_1
(B2) (B3) (B4) (B5)



OPDRACHT 5.2 REGISTERS EN DECODERS (FASE 3)

- Deze laatste uitbreiding (zie hiernaast) is vrij gemakkelijk te realiseren met de schakeling die reeds is opgebouwd op een andere werktafel.
- Zet de twee opgebouwde schakelpanelen naast elkaar en verbind de databus en de voeding.
- Op het rechter schakelpaneel worden op de 74138 de aansluitingen Q_0 naar Q_2 en Q_1 naar Q_3 gebracht.
- We gaan de twee registers van het linker paneel besturen met de 74138, die zich op het rechter paneel bevindt. Hiervoor worden verbonden:
  G_6 van de 7402 linker paneel met Q_0 74138 rechter paneel
  G_8 van de 7402 linker paneel met Q_1 74138 rechter paneel
  Doe dit met blauwe snoeren. Zorg ervoor dat de 7402 van het linker paneel niet meer aangesloten is op de 74138 van hetzelfde paneel.
- Het R/W-commando van de registers op het linker paneel moet los gemaakt worden van de 8-bit-selector op dat paneel en aangesloten worden op het R/W-commando van het rechter paneel.
- U kunt nu informatie opslaan in vier 2-bits registers, die U selecteert met A_0 en A_1 van de decoder.

  A_0 = L en A_1 = L selecteert R_1
  A_0 = H en A_1 = L selecteert R_2
  A_0 = L en A_1 = H selecteert R_3
  A_0 = H en A_1 = H selecteert R_4

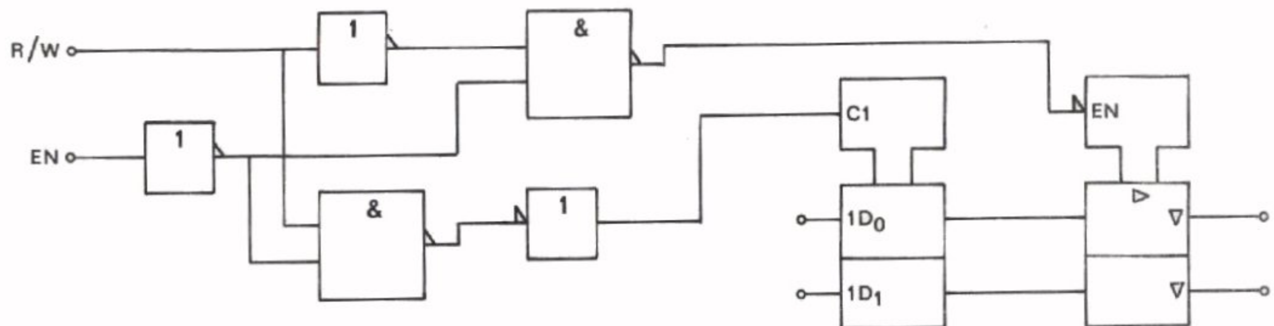
- Probeer
  - in R_1 de data 00 op te slaan
  - in R_2 de data 01 op te slaan
  - in R_3 de data 10 op te slaan
  - in R_4 de data 11 op te slaan

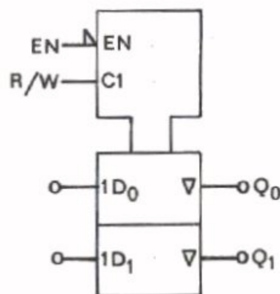
- Schakel nu de voedingsspanning uit en daarna weer aan. De informatie opgeslagen in de registers is verloren.

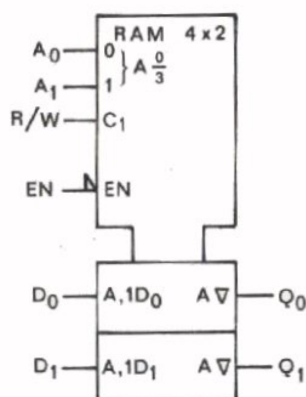
CONCLUSIES

- Met een decoder is het mogelijk het aantal controlelijnen te reduceren, want voor het selecteren van binair naar 1-uit-4-registers zijn maar twee controlelijnen A_0 en A_1 nodig. Voor het selecteren van binair naar 1-uit-8-registers zijn drie lijnen A_0, A_1 en A_2 nodig.
- Met vier registers is het mogelijk vier verschillende data op te slaan en te onthouden, mits de voedingsspanning aangesloten blijft. In de vorige opdracht heeft U in feite een 4 x 2 bits geheugen opgebouwd.

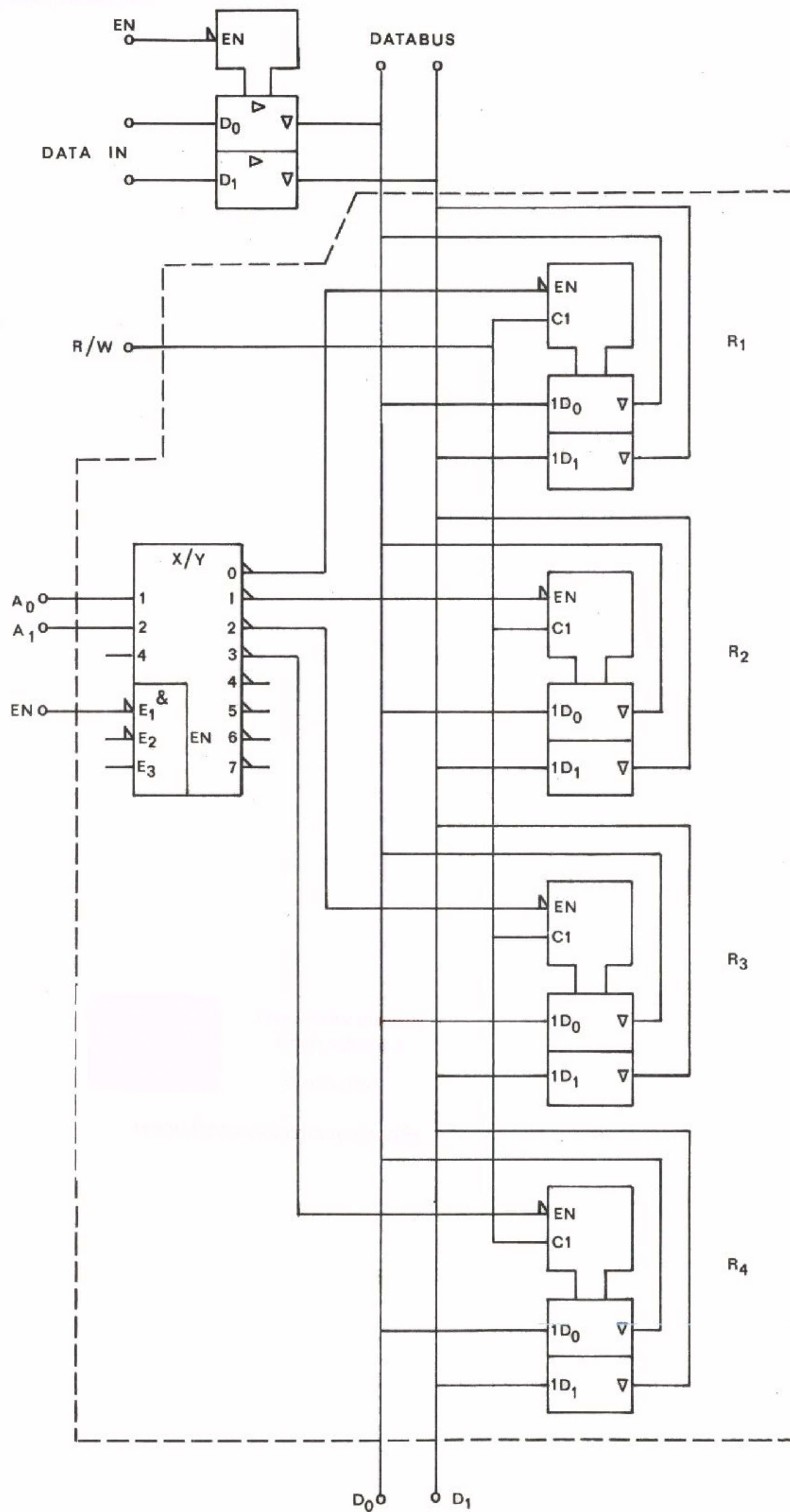
Hieronder hebben we nog eens een 2-bits register getekend met buffer en controlecircuit, zoals U dit tot nu toe verschillende keren hebt gebruikt.





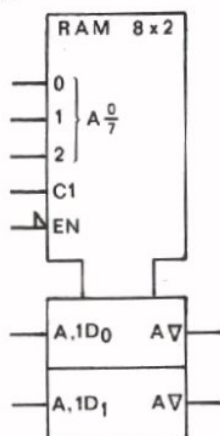


Een dergelijk circuit heeft een symbool gekregen dat hiernaast is weergegeven. Door dit symbool kunnen we het circuit van de vorige opdracht vereenvoudigen, zoals op de volgende pagina is weergegeven. Het circuit binnen de stippellijnen is een geheugen (4 x 2 bits). Aangezien een geheugen een veel voorkomende schakeling is, heeft het ook een symbool gekregen. Dit geheugensymbool is hiernaast getekend. Het woord RAM in het symbool is een afkorting van de Engelse uitdrukking: Random Access Memory. Letterlijk vertaald betekent het een geheugen waarvan alle geheugenplaatsen willekeurig toegankelijk zijn. Als zodanig is deze naam fout, want dat geldt voor de meeste geheugens. Een betere benaming zou zijn Read/Write Memory, een geheugen waarin geschreven en gelezen kan worden door het systeem, waarin het geplaatst is. Het nadeel van de meeste R/W memories is, dat de informatie verloren gaat zodra de voedingsspanning uitgeschakeld wordt.



EN
EN
DATABUS
DATA IN
D0
D1
R/W
EN
C1
1D0
1D1
R1
R2
R3
R4
X/Y
A0
A1
EN
E1
E2
E3
&
0
1
2
3
4
5
6
7
D0
D1

TERMINOLOGIE



Hiernaast ziet U het symbool van een 8 x 2 RAM.

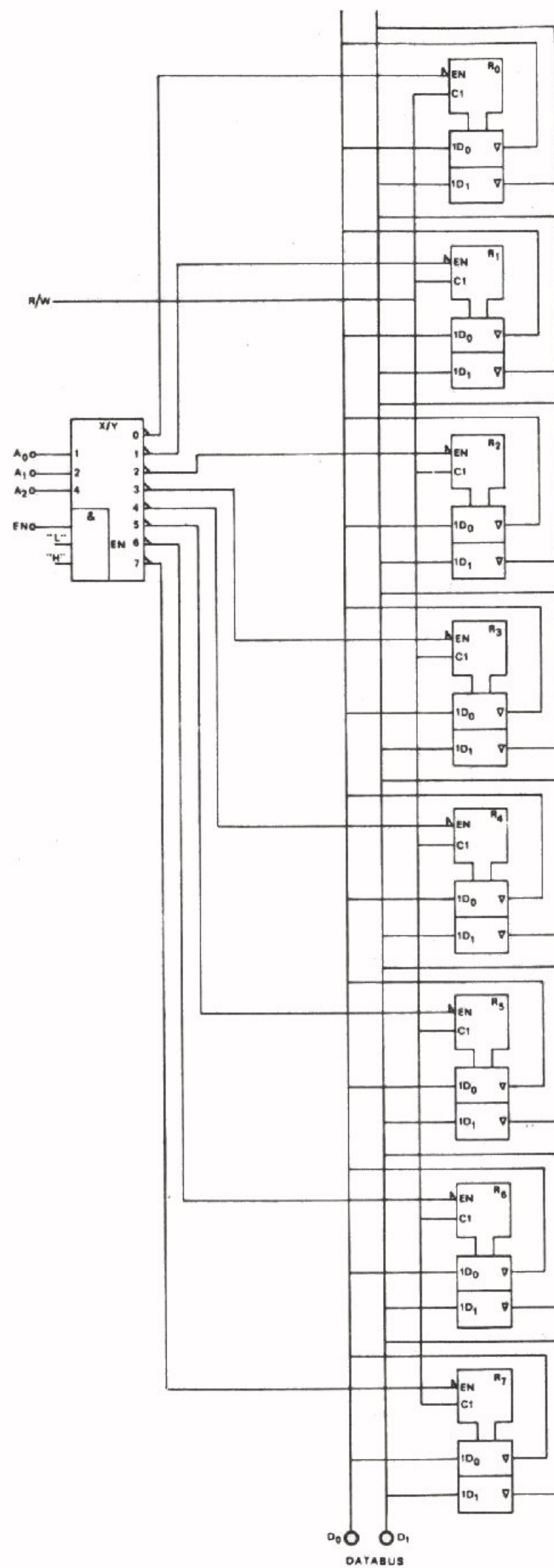
Op de volgende pagina is nog eens weergegeven wat een dergelijke RAM bevat:

- 8 registers met hoogohmige toestand en een gemeenschappelijk R/W commando.
- 1 decoder, in dit geval een 1-uit-8 decoder.

De drie ingangen A_0, A_1 en A_2 dienen om de registers te selecteren, volgens onderstaande tabel.

A_2	A_1	A_0	Geselecteerd register
L	L	L	R_0
L	L	H	R_1
L	H	L	R_2
L	H	H	R_3
H	L	L	R_4
H	L	H	R_5
H	H	L	R_6
H	H	H	R_7

Om bijvoorbeeld data in register R_5 op te slaan, moeten de controle ingangen zijn:
A_2 = H, A_1 = L en A_0 = H.
In het geval van hoog-actief in- en uitgangen komt dit overeen met $A_2 = 1$, $A_1 = 0$ en $A_0 = 1$.
We zeggen dat het getal $A_2A_1A_0 = 101$, register R_5 selecteert, ofwel 101 is het adres van R_5.
De controlelijnen die in een geheugen dient om een bepaald register te selecteren, worden de adreslijnen genoemd. Meestal wordt het Engelse woord ADDRESSBUS gebruikt. De overige, zoals R/W en EN, vormen samen de controlelijnen. De D-uitgangen zijn aangesloten op de datalijnen.


R/W
X/Y
A0
A1
A2
EN
"L"
"H"
&
EN
R0
R1
R2
R3
R4
R5
R6
R7
EN
C1
1D0
1D1
D0
D1
DATABUS

PARALLEL SCHAKELEN VAN RAM'S

In de schakeling van de volgende pagina hebben we vier 8 x 2 RAM's of R/W geheugens parallel geschakeld. Iedere RAM heeft 8 registers, waarin 2-bits data opgeslagen kan worden. De D_0 en D_1 uitgangen zijn parallel aangesloten op de databus. Dit kan, want zodra de EN-ingang op niveau H staat, kan niet uit het geheugen, dus uit één van zijn registers, gelezen worden. U weet, dat in dit geval ook niet in het geheugen, dus in één van zijn registers, geschreven kan worden. We kunnen dus ook alle A_0, A_1 en A_2 lijnen parallel aansluiten. Deze drie lijnen vormen de adresbus. De controlebus wordt gevormd door de EN-ingangen en de gemeenschappelijke R/W lijn van de registers.

SOORTEN GEHEUGENS

De geheugens die we nu bestudeerd hebben zijn halfgeleider geheugens. Ze bestaan in verschillende vormen, bijvoorbeeld:

2606 is een 256 x 4 RAM
82S208 is een 256 x 8 RAM

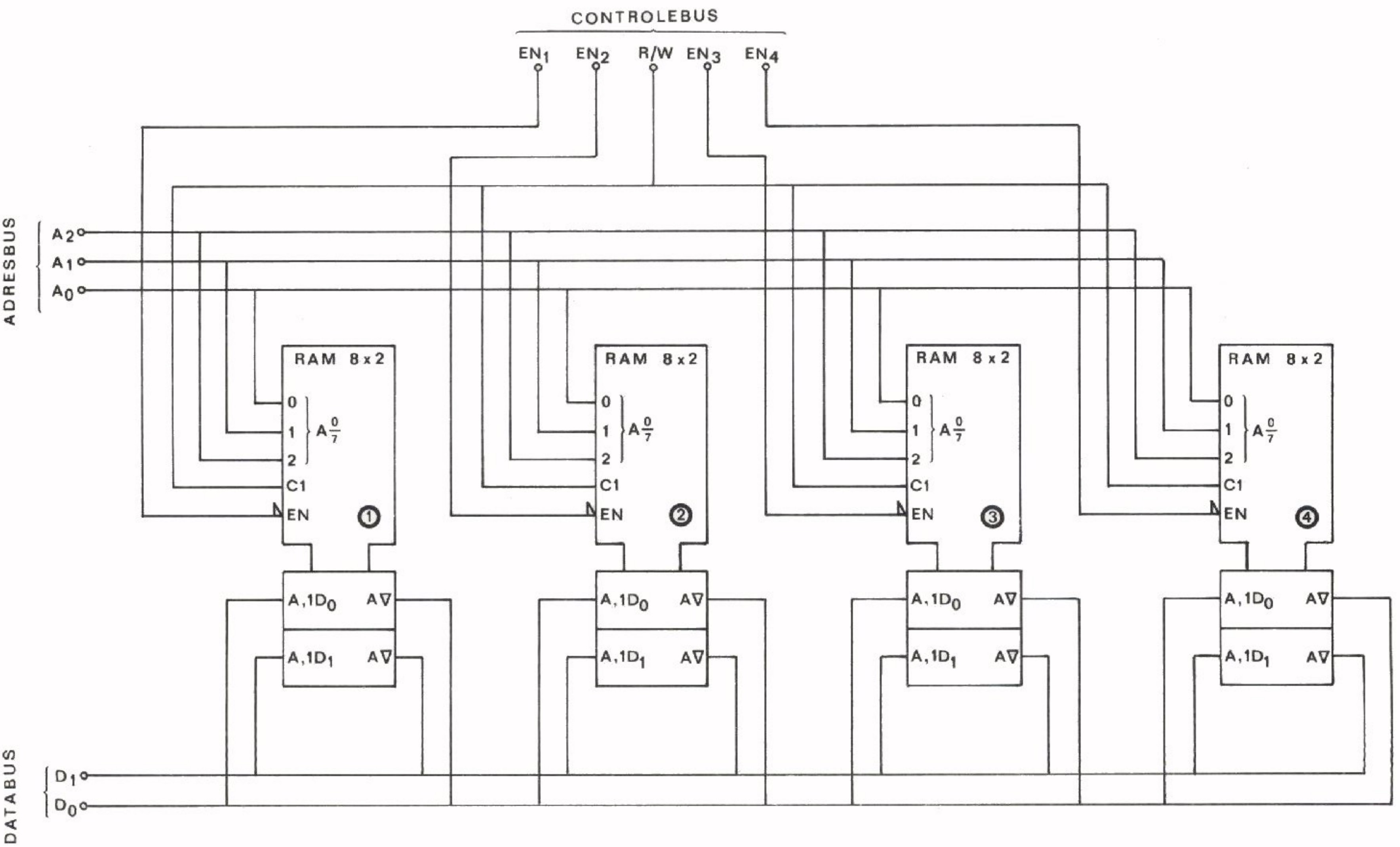
Het aantal bits, dat in een enkel IC opgeslagen kan worden, wordt ieder jaar groter. Zij hebben altijd het nadeel, dat de opgeslagen informatie verloren gaat zodra de voedingsspanning uitgeschakeld wordt.

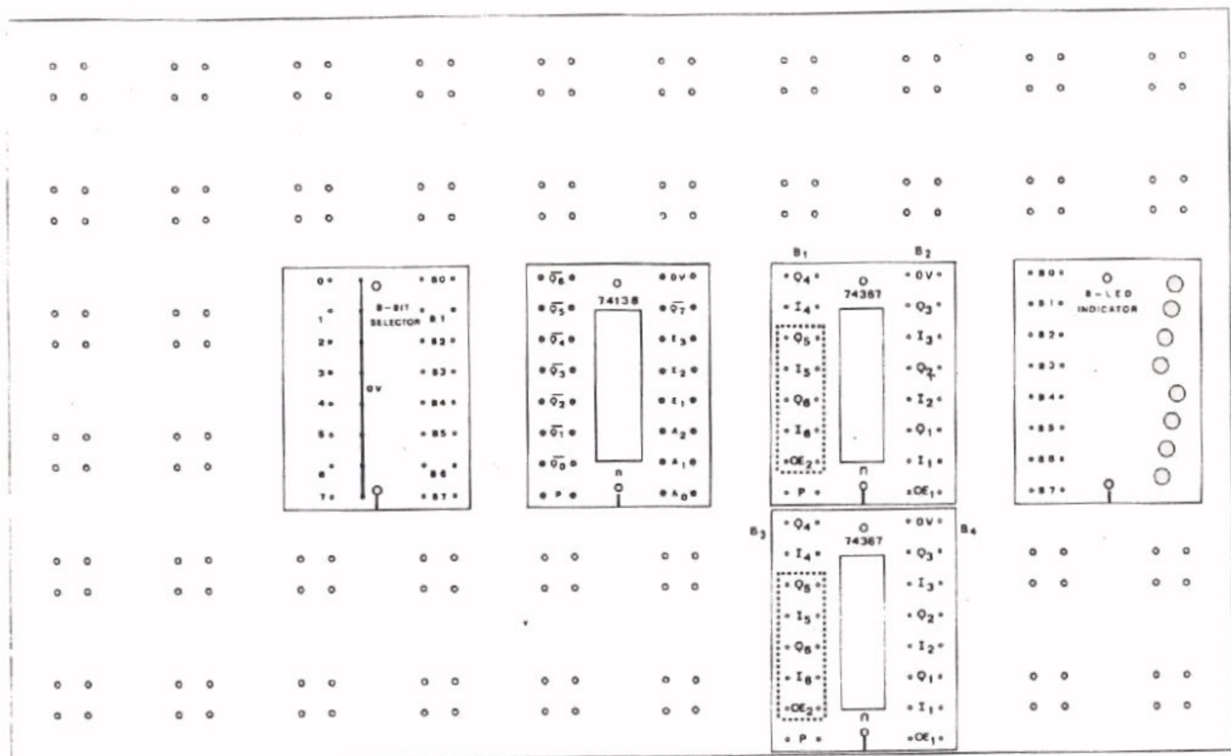
Er bestaan echter ook R/W-geheugens, waarvan de informaties niet verloren gaat bij uitschakeling van de voedingsspanning. Het zijn echter geen halfgeleidersgeheugens en wij zullen ze daarom niet behandelen in deze cursus.

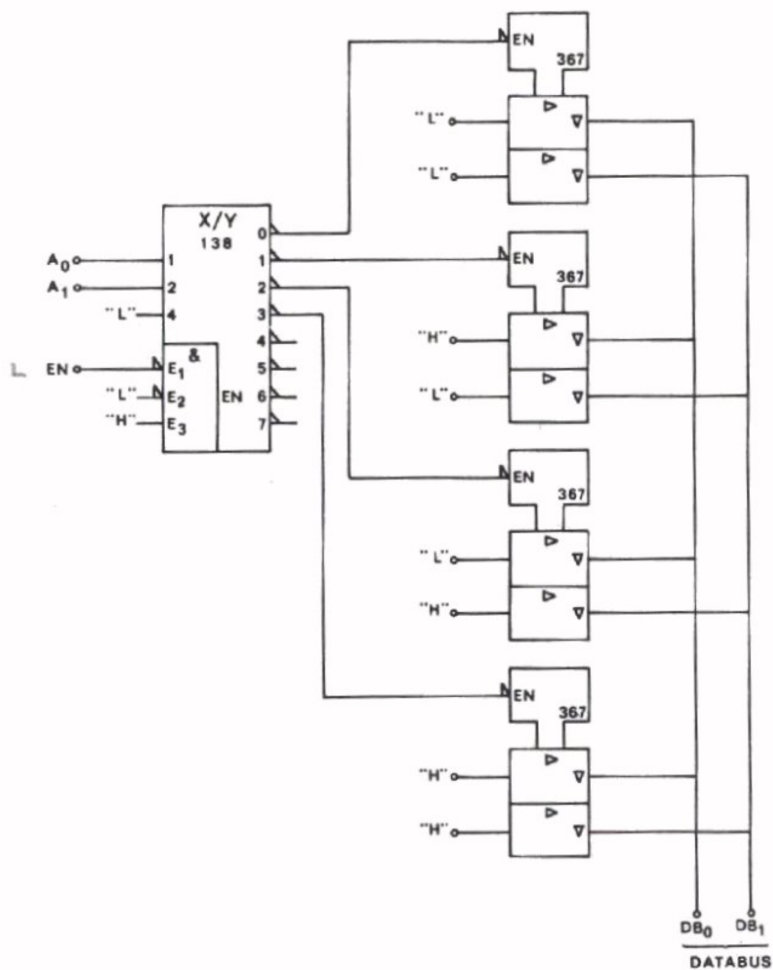
SAMENVATTING

R/W memories, meestal RAM's genoemd, kunnen in elk van hun registers een binaire data onthouden (zolang de voedingsspanning aanwezig is) en beschikbaar te stellen op de databus wanneer er om gevraagd wordt. Het is ook mogelijk op elk gewenst moment de opgeslagen informatie te wijzigen.

Er zijn echter ook geheugens, die uitsluitend gelezen kunnen worden. De fabrikant van dergelijke IC's heeft de informatie in binaire vorm reeds opgeslagen. Deze informatie is niet meer te wijzigen. Dit gaan we in de volgende opdracht bekijken.

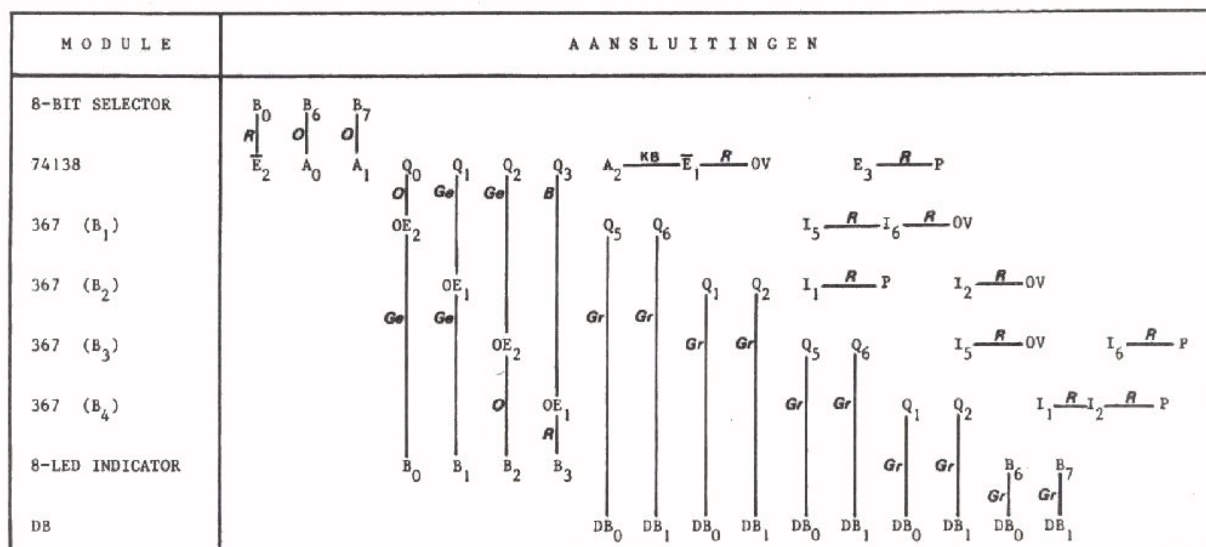

CONTROLEBUS
EN1
EN2
R/W
EN3
EN4
ADRESBUS
A2
A1
A0
RAM 8 x 2
0
1
2
A 0/7
C1
EN
A,1D0
A,1D1
A∇
DATABUS
D1
D0




X/Y
138
A0
A1
"L"
EN
"L"
"H"
E1
E2
E3
&
EN
EN
367
"L"
"H"
DB0
DB1
DATABUS

OPDRACHT 5.3 EEN GEHEUGEN, WAARUIT UITSLUITEND GELEZEN KAN WORDEN

- Plaats de modulen, zoals hiernaast aangegeven.
- Bedraad het circuit van de vorige pagina met behulp van het volgende bedradingsschema:



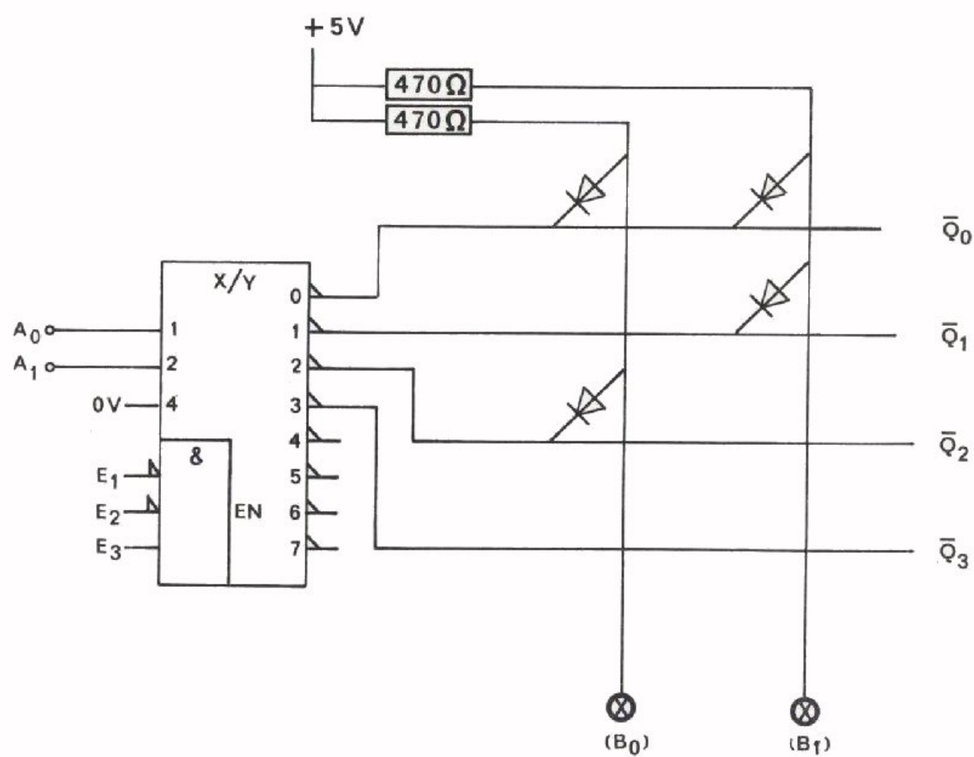
- Vul de volgende tabel in.

A_1	A_0	DB_0	DB_1
L	L		
L	H		
H	L		
H	H		

- Schakel de voedingsspanning even uit en daarna weer aan. Controleer of de vorige tabel nog geldt.

CONCLUSIE

Door middel van de adressen A_0 en A_1 kunnen we de input-data van de buffers selecteren en op de databus schrijven. Deze input-data zelf kunnen niet veranderd worden. Het systeem kan alleen lezen wat er op de verschillende plaatsen staat. Deze schakeling heet een Read Only Memory, afgekort ROM. Dit betekent: een geheugen waaruit men slechts kan lezen. De informatie op de verschillende geheugenplaatsen is er tijdens de fabricage in aangebracht. Het grote voordeel van deze geheugens is, dat de informatie voor altijd bewaard blijft en niet verstoord kan worden.
In de volgende opdracht gaan we een andere soort ROM bekijken.


+5V
470Ω
470Ω
X/Y
A$_0$
A$_1$
0V
1
2
4
&
EN
E$_1$
E$_2$
E$_3$
0
1
2
3
4
5
6
7
$\bar{Q}_0$
$\bar{Q}_1$
$\bar{Q}_2$
$\bar{Q}_3$
(B$_0$)
(B$_1$)

OPDRACHT 5.4 PROGRAMMEERBARE ROM

- Hiernaast ziet U een tekening van het schakelpaneel, waarop onderstaand circuit is gebouwd. Bouw die schakeling na.
- Wat gebeurt er aan de uitgangen B_0 en B_1 van de schakeling, als aan de adresingangen A_0 en A_1 van de 1-uit-8-decoder een adres wordt aangeboden?

  Vul hiervoor onderstaande tabel in:

A_1	A_0	B_1	B_0
L	L		
L	H		
H	L		
H	H		

- Vergelijk de uitkomst met de plaats van de diodes.
- Wat is de invloed op de resultaten, als één diode wordt verwijderd?

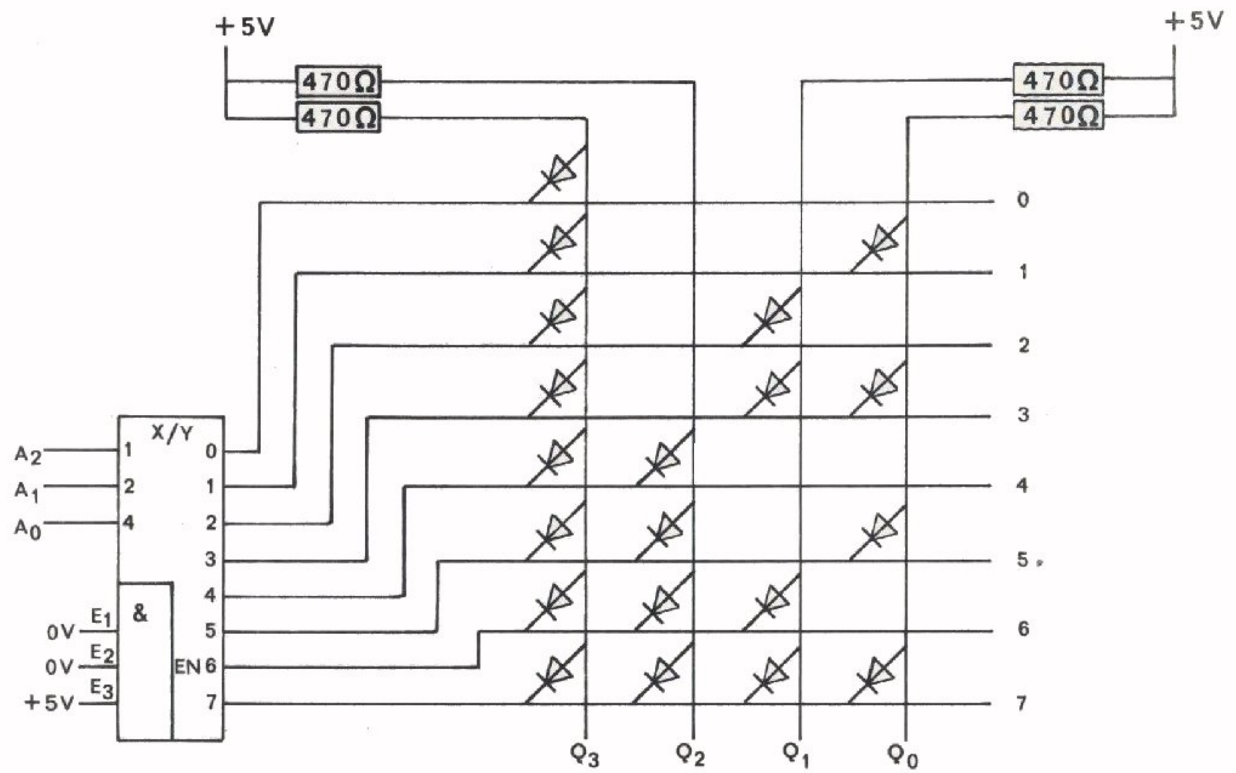
CONCLUSIE

De schakeling gedraagt zich als een geheugen, dat door het systeem alleen maar "gelezen" kan worden, dus als een ROM. Heeft men zelf de mogelijkheid om de diodes aan te brengen of te verwijderen, dan spreken we van een Programmable Read Only Memory, afgekort PROM. Dit betekent: Programmeerbaar geheugen dat slechts gelezen kan worden.

DE PROM

Het circuit hiernaast is een uitbreiding van het circuit, dat we gebruikt hebben tijdens de vorige opdracht. Het circuit bevat een decoder en een matrix. Deze matrix is een raamwerk van een aantal horizontale geleiders verbonden via diodes met een aantal verticale geleiders. Het kan op eenvoudige wijze veranderd worden door diodes er uit te halen of er in te plaatsen. Een geselecteerde horizontale lijn bepaalt via de diodes de niveaus van de uitgangen. Deze PROM kan door de gebruiker zelf op vrij eenvoudige wijze geprogrammeerd worden.

In een PROM, zoals bijvoorbeeld de 82S123 zijn op alle geleider-kruisingen diodes aangebracht. Het IC wordt geprogrammeerd door een of meerdere diodes door een relatief grote stroom te onderbreken. Hierdoor wordt die verbinding in de matrix verbroken. De uitgangsspanning zal dus H (hoog) blijven, als de lijn met de kapotte diode weer wordt geselecteerd.


+5V
470Ω
470Ω
+5V
470Ω
470Ω
0
1
2
3
4
5
6
7
X/Y
A2
A1
A0
1
2
4
0
1
2
3
4
5
EN 6
7
&
0V E1
0V E2
+5V E3
Q3
Q2
Q1
Q0

6

TALSTELSELS

INLEIDING

In deel D van de cursus Bedrijfselektronica (lessen D18 en D19) hebben we uitvoerig gesproken over het binaire, het octale en het hexadecimale talstelsel. We hebben onder andere geleerd twee binaire getallen op te tellen, af te trekken en te vermenigvuldigen. Hieruit werden de begrippen carry en borrow voor U duidelijk.

In dit hoofdstuk gaan we opnieuw aandacht besteden aan het binaire talstelsel. Enkele nieuwe punten die in het bijzonder voor microcomputers van belang zijn, worden uitvoerig behandeld, zoals bijvoorbeeld het rekenen met negatieve getallen en het rekenen met behulp van de BCD code.

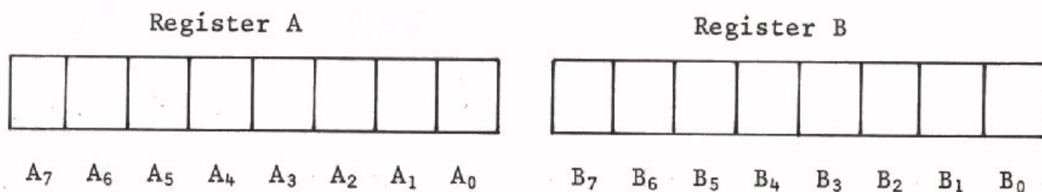
WOORDLENGTE

Wanneer wij getallen opschrijven, werken wij vaak met getallen van zeer uiteenlopende lengten. Rekenmachines daarentegen werken meestal met getallen die uit een vast aantal symbolen samengesteld zijn. Dit aantal symbolen noemt men de woordlengte. De woordlengten die in computers worden toegepast, lopen nogal uiteen. Veel gebruikte lengten zijn 8, 12, 16, 18, 24, 32, 48 en 64 bits. Veel moderne microcomputers maken gebruik van woorden met een lengte van 8 bits. Voor speciale toepassingen zijn er ook wel microprocessoren met een woordlengte van 4 bits. Deze worden speciaal gebruikt voor het rekenen met decimale getallen.

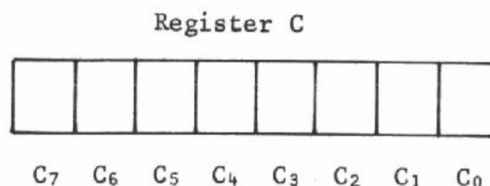
Heeft een machine maar een beperkte woordlengte, dan zal het toch vaak nodig zijn ook met grotere woordlengten te werken. Men deelt dan het grotere woord in een aantal woorden op ter grootte van het standaardwoord van de rekenmachine. Hier komen we later nog op terug.

De beperking in woordlengte van een machine kan soms problemen opleveren. We willen bijvoorbeeld twee getallen optellen die elk 8 bits breed zijn. Alvorens de getallen op te tellen worden ze opgeslagen in registers.

De opbergruimte van de registers kunnen we als volgt schetsen.



Als de machine nu de optelling van A en B uitvoert gaat hij het resultaat, bijvoorbeeld, in een derde register opslaan. Dit register, C, is ook 8 bits breed.



Tijdens de optelling kan hier en daar een carry optreden die naar de nevenstaande bit met hoger gewicht overgeheveld moet worden. Het is van belang daarbij twee categorieën van carries te onderscheiden.

1. carries die binnen register C overgeheveld worden. Hiervan merkt men buiten dit register dus niets.

2. een carry die door de optelling van A_7 en B_7 optreedt. Er ontstaat dan in C_7 een carry die buiten de lengte van dit register valt. Deze carry moet naar een ander register overgebracht worden, of men is hem kwijt. Van deze carry categorie merkt men buiten register C wél wat en voor hem moeten extra maatregelen getroffen worden.

Dezelfde problemen kunnen ook voorkomen tijdens het aftrekken van twee getallen, met name als een borrow optreedt.

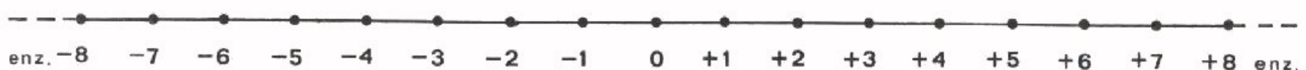
DE GETALLEN-RECHTE

Bij het rekenen in het *decimale talstelsel* heeft men te doen met OPTELLEN en met AFTREKKEN. Dit zijn verschillende handelingen. Bij het optellen krijgt men af en toe te maken met carries; bij het aftrekken krijgt men daarentegen te maken met borrows.

Als men een getal aftrekt van een kleiner ander getal, geeft de uitkomst weer hoeveel men tekort komt. Rekenkundig geeft men dit aan door een "-" teken voor het getal te zetten. Men noemt dat getal een NEGATIEF GETAL.

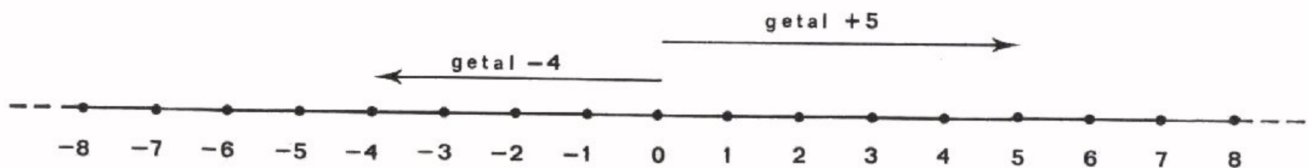
Men kan negatieve en positieve getallen uitzetten langs een rechte lijn, de zogenaamde GETALLEN-RECHTE. Deze is hieronder voor een deel (in de buurt van 0) getekend.

de getallen-rechte

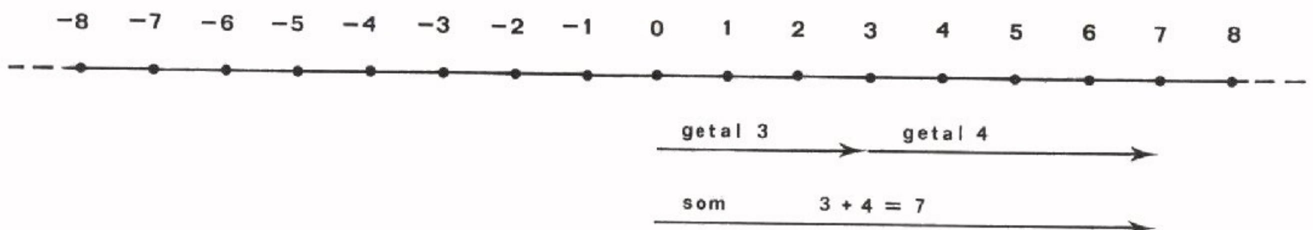


Uit de figuur blijkt dat de getallen van 0 uitgaande in beide richtingen aangroeien volgens 1, 2, 3, 4, enz. Daarbij worden voor de negatieve getallen - tekens geplaatst, terwijl voor de gewone getallen (die men wel de positieve getallen noemt) + tekens komen. De + tekens mag men weglaten.

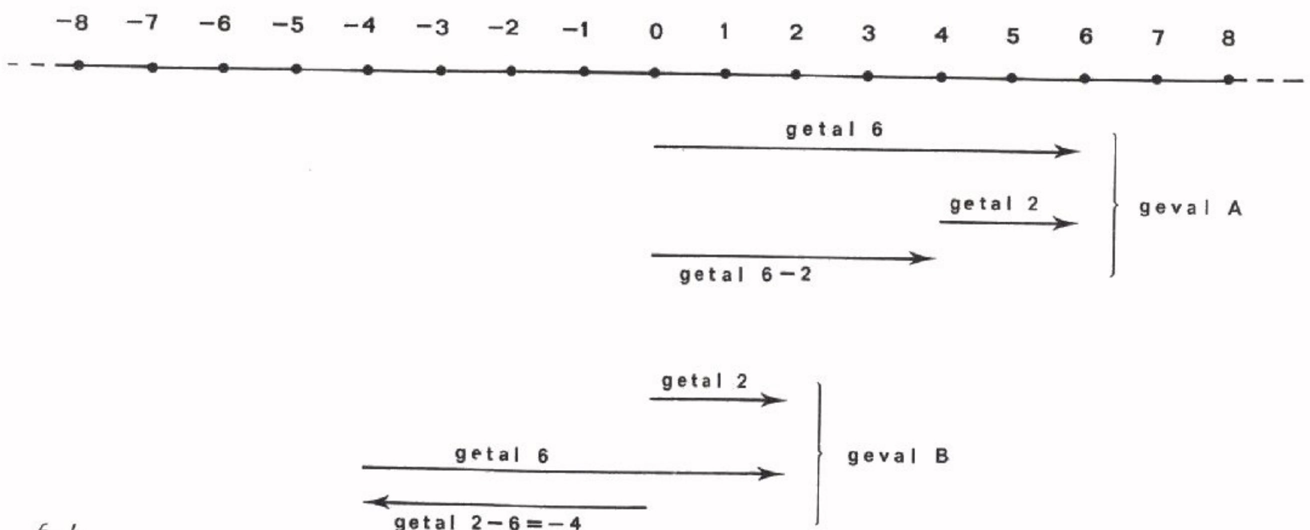
Men kan de getallen ook voorstellen door pijlen of vectoren. De lengte van de vector is dan een maat voor de grootte van het getal. Wijst de vector naar rechts, dan stelt hij een positief getal voor. Wijst hij naar links, dan stelt hij een negatief getal voor, zie hieronder.



Het *optellen* van twee getallen komt er nu op neer, dat men de twee vectoren van deze getallen achter elkaar zet en nagaat hoe groot de totale lengte wordt.



Het *aftrekken* van twee getallen komt er op neer, dat men het verschil in lengte van deze getallen bepaalt.

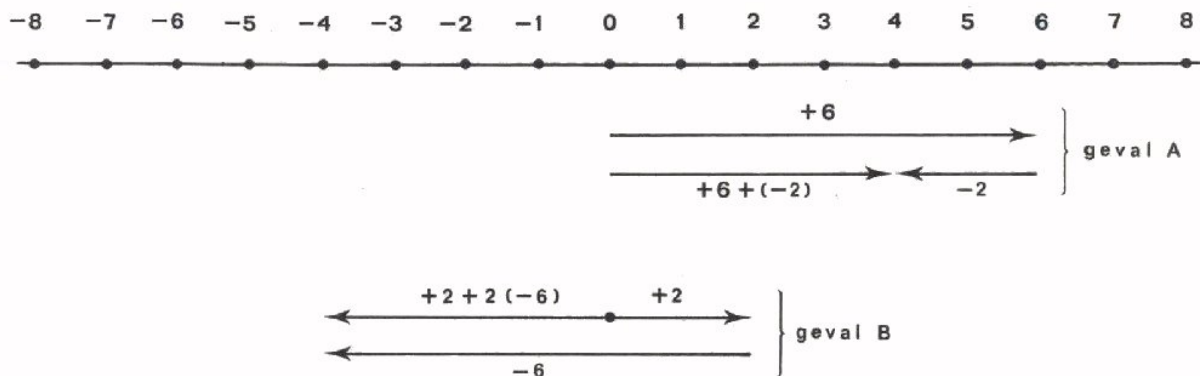


Trekt men een getal van een groter getal af, dan is het verschil positief (geval A).

Trekt men een getal van een kleiner getal af, dan is er een tekort zodat het verschil negatief is (geval B).

De algebra leert: +2 - (+6) = +2 + (-6).

Onder woorden gebracht is dit: een positief getal G ergens van aftrekken komt op hetzelfde neer als het negatieve getal G er bij optellen. Dit valt mooi te demonstreren met onze vectorvoorstelling van getallen.



We hebben in het laatste figuur het aftrekken eigenlijk omgezet in een optelhandeling.

Het optellen en het aftrekken kan nu gebeuren met éénzelfde optelhandeling. Deze bestaat daarin, dat men de getalvectoren eerst achter elkaar aanzet; vervolgens vindt men de somvector door het achtereinde van de eerste getalvector a door te verbinden met de pijlpunt van de tweede getalvector.



We hebben met behulp van de vectorvoorstelling van getallen het optellen en aftrekken tot éénzelfde handeling terug kunnen brengen. We gaan nu zien dat door de notatie van negatieve getallen in het binaire stelsel handig te kiezen, het ook mogelijk is met één zelfde handelwijze twee getallen zowel op te tellen als af te trekken.

NEGATIEVE GETALLEN IN HET BINAIRE TALSTELSEL

In de rekenkunde komen veel getallen voor die negatief zijn. Het middel om dit aan te geven, is eenvoudig het plaatsen van een minteken voor het desbetreffende getal. Positieve getallen worden wel met behulp van een + teken aangegeven; meestal laat men dit teken weg.

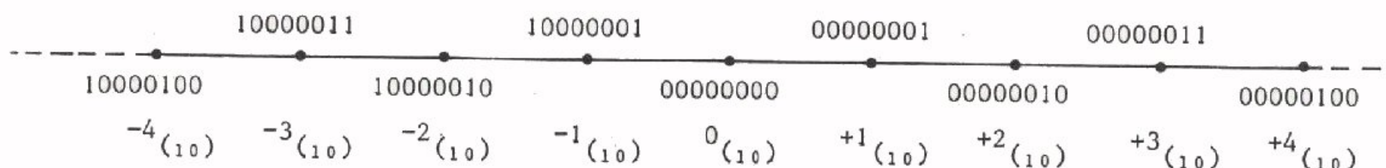
Met een digitale rekenmachine kan men uitsluitend nullen en énen verwerken; + tekens en - tekens kan men niet apart ook nog verwerken. Er zit dus niets anders op dan een bepaalde bit voor het teken van een getal te reserveren en dit teken dan door een 0, resp. 1, vast te leggen.

Men heeft afgesproken om bij de binaire notatie van een getal altijd DE MEEST LINKSE BIT voor HET TEKEN te reserveren: is deze bit 0, dan is het getal positief; is de bit 1, dan is het getal negatief.

Verder moet men bedenken, dat een te verwerken getal altijd uit een beperkt aantal bits is samengesteld. Wordt het getal bijvoorbeeld in een 8 bits-register opgeslagen, dan is één bit voor het teken gereserveerd en schieten er zeven bits over voor de cijfers van het binaire getal (zie figuur).



We zouden er nu over kunnen denken om de negatieve binaire getallen net zo weer te geven als het gebruikelijk is voor decimale getallen. Dat wil zeggen, van 0 afgaand de negatieve getallen net zo laten oplopen als de positieve getallen (zie de getallen-rechte volgens figuur hieronder).



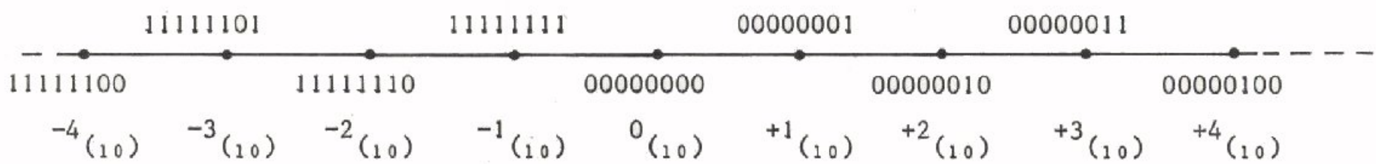
We hebben ervoor gekozen $0_{(10)}$ bij de positieve getallenreeks te laten horen (het had ook andersom gekund).

Leggen we de negatieve binaire getallen vast volgens de figuur hiernaast, dan zitten we met het nadeel dat optellen en aftrekken verschillende handelingen blijven.
Immers, tellen we een positief en een negatief binair getal op, dan is de uitkomst niet goed.

Voorbeeld:

	0 0000010	⟶	$+2_{(10)}$
	1 0000100	⟶	$-4_{(10)}$
+	1 0000110	⟶	$-6_{(10)}$

We kunnen echter ook de notatie van de positieve binaire getallen normaal blijven aanhouden (opklimmend van 0 afgaand), maar de notatie van de negatieve getallen laten afnemen van 0 afgaand. Dit is hieronder weergegeven.



We gaan nu weer eens kijken wat optellen van een positief en een negatief binair getal (op dezelfde wijze als het optellen van twee positieve getallen uitgevoerd) nu oplevert. We noemen de absolute waarden van deze getallen A en B.

Voorbeelden:

+A =	0 0000010	⟶	$+2_{(10)}$	
−B =	1 1111100	⟶	$-4_{(10)}$	A < B
(+A) + (−B) =	1 1111110	⟶	$-2_{(10)}$	

+A =	0 0000100	⟶	$+4_{(10)}$	
−B =	1 1111110	⟶	$-2_{(10)}$	A > B
(+A) + (−B) =	(1) 0 0000010	⟶	$+2_{(10)}$	

+A =	0 0000100	⟶	$+4_{(10)}$	
−B =	1 1111100	⟶	$-4_{(10)}$	A = B
(+A) + (−B) =	(1) 0 0000000	⟶	$0_{(10)}$	

We hebben drie uiteenlopende voorbeelden genomen:

- het geval dat A kleiner is dan B.
- het geval dat A groter is dan B.
- het geval dat A gelijk is aan B.

Het blijkt dat in al deze gevallen de uitkomst juist is.
Verder verdwijnt in het 2e en 3e geval een bit 1 uit het register (deze hebben we met ① weergegeven). De achtergebleven uitkomsten zijn echter juist.

In binaire rekenkunde houdt men de notatie van negatieve binaire getallen op de laatste manier aan. Dit levert dan volgende voordelen op:

a. Uit de meest linkse bit volgt onmiddellijk of een getal positief hetzij negatief is.

b. Men kan positieve en negatieve getallen (en ook negatieve en negatieve getallen) op dezelfde wijze optellen als positieve bij positieve getallen worden opgeteld.

c. Het aftrekken van getallen vereist geen nieuwe handeling: het aftrekken van een getal B wordt uitgevoerd als het optellen van een getal -B.

In een 8 bits-register kan één van 2^7 uiteenlopende binaire getallen worden opgeslagen die links met 0 beginnen (de positieve getallen).

Verder kan in dat register ook één van 2^7 uiteenlopende binaire getallen worden opgeslagen die links met een 1 beginnen (de negatieve getallen).

Onder de positieve getallen bevindt zich echter de $0_{(10)}$ in de vorm van 0 0 0 0 0 0 0 0, die eigenlijk noch positief, noch negatief te noemen is. In feite kan men dus één van $2^7 - 1$ positieve binaire getallen en één van 2^7 negatieve binaire getallen opslaan. De reeks negatieve getallen bevat blijkbaar een getal meer dan de reeks positieve getallen.

2-COMPLEMENT IN HET BINAIRE TALSTELSEL

Hoe zet men een positief binair getal om in een evengroot negatief binair getal, en omgekeerd? We gaan dit behandelen aan de hand van een voorbeeld. Daarna verstrekken we het toegepaste recept, dat geheel in het algemeen blijkt op te gaan.

Eerst de omzetting van positief naar negatief.

Voorbeeld:

$+4_{(10)}$ =	0 0000100	
	1 1111011	inverteren
	0 0000001	$+1_{(10)}$ bij optellen
+	1 1111100	$= -4_{(10)}$

Als we het positieve getal inverteren, krijgen we een negatief getal waarvan de absolute waarde $1_{(10)}$ te groot is. Daarom moeten we die absolute waarde met $1_{(10)}$ verminderen. Dit is te doen door er nog eens $+1_{(10)}$ = 0 0000001 bij op te tellen.

- Het recept voor de omzetting van een positief binair getal in een negatief binair getal met een evengrote absolute waarde is dus:
  - Eerst het positieve binaire getal inverteren. Men krijgt dan het zogenaamde 1-complement.
  - Daarna $+1_{(10)}$ in binaire vorm bij het 1-complement optellen. De verkregen uitkomst noemt men het 2-complement en dit is het gewenste negatieve binaire getal.

Vervolgens de omzetting van negatief naar positief.

Voorbeeld:

$-3_{(10)}$ =	1 1111101	
	0 0000001	er $+1_{(10)}$ van aftrekken
−	1 1111100	
	0 0000011	inverteren $= +3_{(10)}$

- Hier is de gevolgde procedure net andersom: eerst $+1_{(10)}$ er van aftrekken en vervolgens het verkregen binaire getal inverteren.

TALSTELSEL OVERZICHT

Hieronder treft U een overzicht aan van de talstelsels die we óf in Deel D, en in dit hoofdstuk bestudeerd hebben.

B C D - stelsel	hexa-decim.	deci-maal	binair talstelsel
0 0 0 1 0 1 1 0	1 0	1 6	0 0 0 1 0 0 0 0
0 0 0 1 0 1 0 1	F	1 5	0 0 0 0 1 1 1 1
0 0 0 1 0 1 0 0	E	1 4	0 0 0 0 1 1 1 0
0 0 0 1 0 0 1 1	D	1 3	0 0 0 0 1 1 0 1
0 0 0 1 0 0 1 0	C	1 2	0 0 0 0 1 1 0 0
0 0 0 1 0 0 0 1	B	1 1	0 0 0 0 1 0 1 1
0 0 0 1 0 0 0 0	A	1 0	0 0 0 0 1 0 1 0
0 0 0 0 1 0 0 1	9	9	0 0 0 0 1 0 0 1
0 0 0 0 1 0 0 0	8	8	0 0 0 0 1 0 0 0
0 0 0 0 0 1 1 1	7	7	0 0 0 0 0 1 1 1
0 0 0 0 0 1 1 0	6	6	0 0 0 0 0 1 1 0
0 0 0 0 0 1 0 1	5	5	0 0 0 0 0 1 0 1
0 0 0 0 0 1 0 0	4	4	0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 1	3	3	0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 0	2	2	0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1	1	1	0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0	0	0	0 0 0 0 0 0 0 0
		- 1	1 1 1 1 1 1 1 1
		- 2	1 1 1 1 1 1 1 0
		- 3	1 1 1 1 1 1 0 1
		- 4	1 1 1 1 1 1 0 0
		- 5	1 1 1 1 1 0 1 1
		- 6	1 1 1 1 1 0 1 0
		- 7	1 1 1 1 1 0 0 1
		- 8	1 1 1 1 1 0 0 0
		- 9	1 1 1 1 0 1 1 1

2e tetrade 1e tetrade

tekenbits

binnen 8-bits register

OVERFLOW

In het vorige gedeelte, waar het 2-complement getal behandeld is, waren n plaatsen beschikbaar om symbolen in een woord te plaatsen. Hierdoor konden in totaal 2^n verschillende getallen worden gedefinieerd. Deze getallen waren voor de helft positief, voor de andere helft negatief, hetgeen betekent dat er de 0 en 2^n-1 positieve getallen aanwezig kunnen zijn. Dit heeft tot gevolg dat het grootste positieve getal $2^{n-1}-1$ kan zijn. Dat was in het vorige hoofdstuk ook duidelijk te zien. Door de beperkte woordlengte zal het niet mogelijk zijn getallen van onbeperkte grootte weer te geven. Men kan zich nu afvragen, als twee positieve getallen worden opgesteld, wat zal dan de som zijn? Als we een woordbreedte van 4 bits veronderstellen zal deze som de waarde 7 niet mogen overtreffen; 7 is immers het grootste positieve getal dat bij 4 bits voorgesteld kan worden. Stel als voorbeeld dat de getallen 5 en 6 bij elkaar worden opgeteld. Zie hieronder.

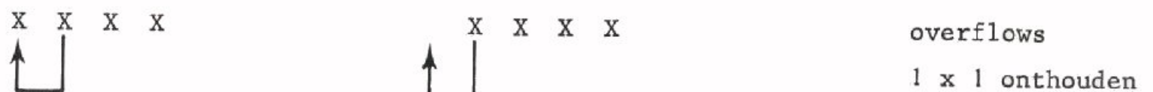
```
+5 = 0 1 0 1
+6 = 0 1 1 0
     -------- +
     1 0 1 1 = -5
     8 4 2 1
          1 x 1 onthouden
```

De som is nu -5. Dit is uiteraard foutief. Welke fout is hierbij gemaakt? Tijdens de optelling is op de derde plaats een overdracht van een 1 ontstaan, die daarna op de vierde plaats is gezet. Dit betekent een verlies van een 8, en bovendien een verkeerde interpretatie van de overige getallen. De totale binaire som is nog volkomen correct, namelijk 11, maar door de afspraak over de afbeelding van negatieve getallen, is deze representatie van 11 niet geldig. Het overlopen van de 1 tussen de beide plaatsen noemt men een *overflow*. Deze overflow kan men vaststellen doordat bij een optelling van 2 positieve of 2 negatieve getallen de meest significante bit verandert. In het bovenstaande voorbeeld waren de beide meest significante bits nullen, en in het resultaat verscheen een 1. Dat is een aanwijzing dat er een getal is ontstaan dat boven de rekencapaciteit van het rekenkundige orgaan uitgaat. Hetzelfde kan optreden bij het optellen van twee negatieve getallen, zoals in onderstaand voorbeeld:

```
-5 =   1 0 1 1
-6 =   1 0 1 0
     --------- +
   1 | 0 1 0 1

     1 x 1 onthouden
```

In dit voorbeeld is de meest significante bit eveneens gewijzigd, terwijl bovendien een 1 buiten het rekenbereik van de machine is gekomen! Dit gezamenlijk wijst erop dat er sprake is van een overflow. In het algemeen geldt dat er een overflow bij de optelling, resp. aftrekking heeft plaats gehad indien er transport van een 1 plaats heeft van de op één na meest significante bit naar de meest significante bit, dan wel van de meest significante bit naar de plaats die niet meer aanwezig is:



Bij het optellen van negatieve getallen zal er altijd een 1 buiten het rekengebied getransporteerd worden. Blijven de negatieve getallen echter binnen het bereik van de machine, dan zal er ook van de op één na meest significante plaats naar de meest significante bit een overdracht van een 1 optreden, zoals uit onderstaand voorbeeld blijkt:

```
 -3 =   1 1 0 1
 -4 =   1 1 0 0
      ─────────── +
  1   | 1 0 0 1 = -7

  2 x 1 onthouden
```

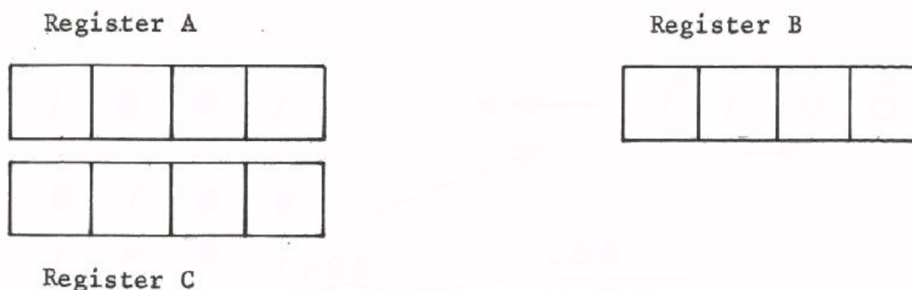
Het optreden van twee overdrachten of het optreden van geen enkele overdracht duidt er dus op dat er géén overflow plaats heeft. Om dit bij gemengde getallen te illustreren zijn hieronder nog twee voorbeelden gegeven:

```
 +4 =   0 1 0 0              +3 = 0 0 1 1
 -3 =   1 1 0 1              -4 = 1 1 0 0
      ────────── +                ──────── +
  1     0 0 0 1 = +1              1 1 1 1 = -1

  2 x 1 onthouden            géén onthouden
```

VERMENIGVULDIGEN VAN BINAIRE GETALLEN

Het vermenigvuldigen in een rekenmachine kan men uitvoeren met behulp van drie registers. We zullen aannemen dat we beschikken over drie 4-bits registers zoals ze hieronder zijn geschetst.



In het volgende voorbeeld gaan we twee 4-bits getallen met elkaar vermenigvuldigen. We gaan er van uit dat het twee positieve getallen zijn; de notatie voor negatieve getallen wordt dus niet toegepast. De gekozen getallen zijn:

$$A = 1\ 0\ 0\ 1_{(2)} = 9_{(10)}$$

$$B = 1\ 1\ 0\ 1_{(2)} = 13_{(10)}$$

We gaan eerst het getal A in register A en getal B in register B opslaan. Ook moeten we ervoor zorgen dat register C alleen maar nullen bevat.



We gaan de getallen A en B met elkaar vermenigvuldigen door herhaald schuiven en optellen. Dit zal hieronder stap voor stap worden uitgelegd.

1. We nemen eerst de LSB van getal B. In ons geval is dit een "1". Nu wordt getal A met "1" vermenigvuldigd en het resultaat bij de inhoud van register C opgeteld.
   De regels voor het vermenigvuldigen van binaire getallen zijn bijzonder eenvoudig:

$$A \times 0 = 0$$

$$A \times 1 = A$$

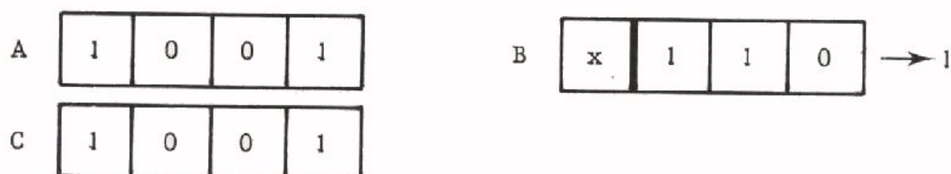
In ons voorbeeld hebben we dus, als eerste stap:

1 0 0 1 x 1 = 1 0 0 1

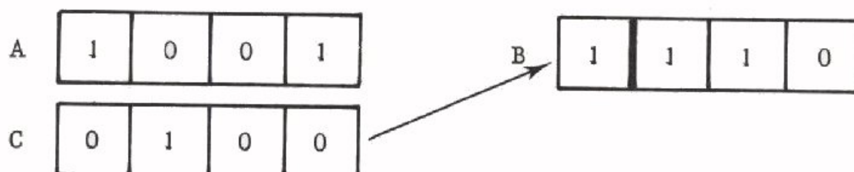
Dit resultaat wordt bij de inhoud van C opgeteld.



2. De LSB van getal B is nu niet meer interessant voor de verdere berekeningen. We kunnen hem uit register B laten schuiven zoals hieronder aangegeven.



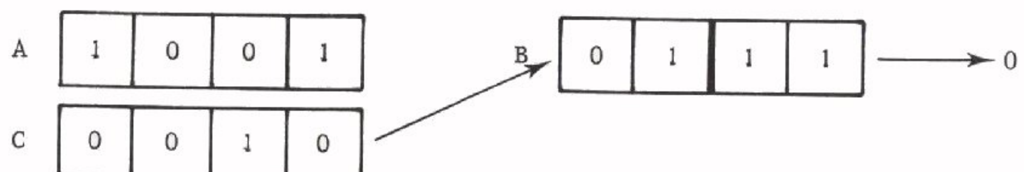
Na die verschuiving zijn we de LSB van getal B kwijt en er is een positie in register B vrijgekomen. Deze vrije positie kunnen we goed benutten aangezien het product van twee getallen een grotere lengte heeft dan die van elk van de twee operanden afzonderlijk. We gaan register C een positie naar rechts schuiven en de laatste bit die eruit valt in de vrije positie van register B plaatsen.



3. Nu bekijken we de nieuwe LSB van register B. Het is een "0".

1 0 0 1 x 0 = 0 0 0 0

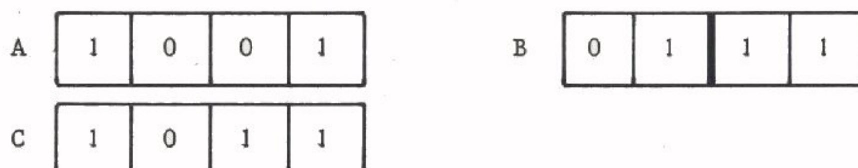
We hoeven niets op te tellen bij de inhoud van register C. Het is voldoende om de inhoud van registers B en C een positie naar rechts te schuiven.



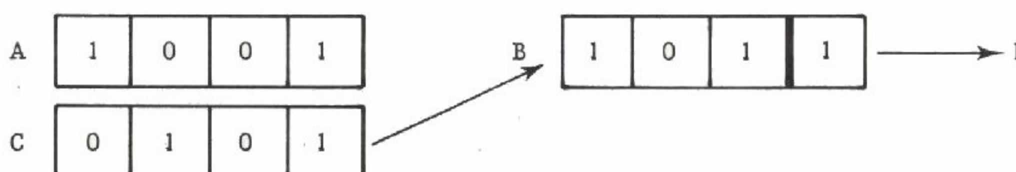
4. De LSB van register B is nu weer een "1". We hebben eerst

1 0 0 1 x 1 = 1 0 0 1

Dit resultaat gaan we nu bij het getal dat in register C aanwezig is optellen.



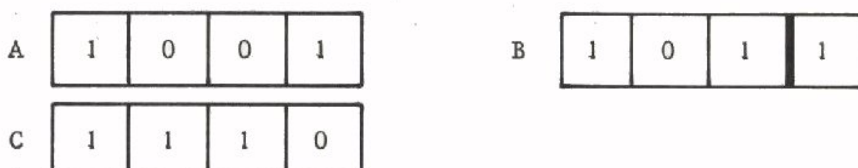
En daarna nog eens schuiven:



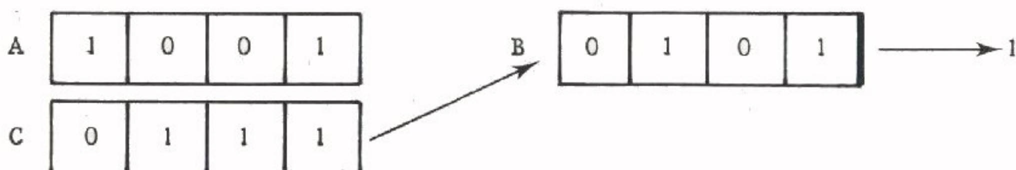
5. In register B blijft de MSB van het oorspronkelijke getal B nog over. Het is een "1" en we gaan hem behandelen, zoals in stap 4: eerst

1 0 0 1 x 1 = 1 0 0 1

bij de inhoud van register C optellen:



En dan schuiven:



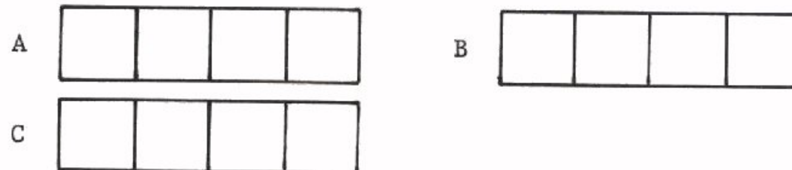
We constateren dat de operand die in stap 1 in register B opgeslagen was nu verloren is en dat het product opgeslagen is in registers C en B.

$$1\,0\,0\,1_{(2)} \times 1\,1\,0\,1_{(2)} = 0\,1\,1\,1\,0\,1\,0\,1_{(2)}$$

$$9_{(10)} \times 1\,3_{(10)} = 1\,1\,7_{(10)} .$$

DELEN VAN BINAIRE GETALLEN

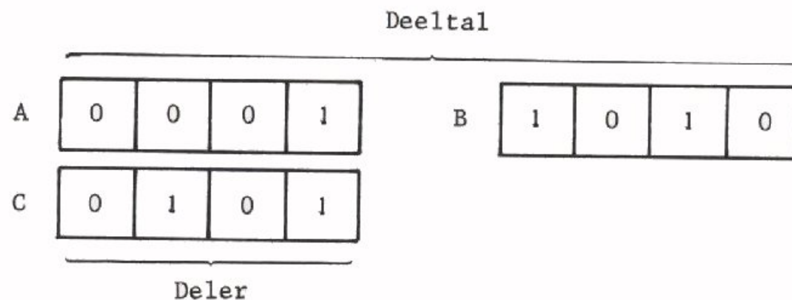
Voor het delen van twee binaire getallen gebruiken we ook drie registers zoals hieronder geschetst. We gaan er weer vanuit dat we beschikken over 4-bits registers.



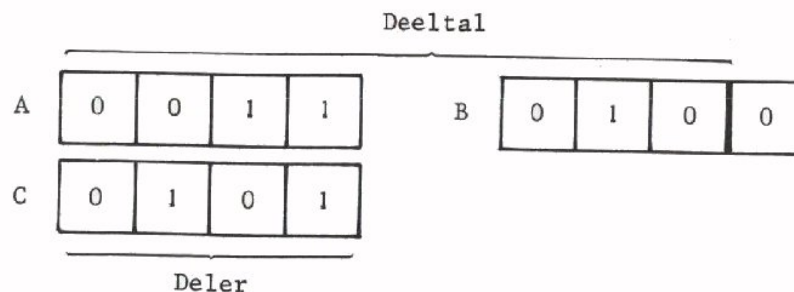
Het delen van twee binaire getallen gebeurt door herhaald schuiven en aftrekken. Dit gaan we stap voor stap bekijken met behulp van het volgende voorbeeld:

$$\frac{1\,1\,0\,1\,0_{(2)}}{0\,1\,0\,1_{(2)}} = \frac{26_{(10)}}{5_{(10)}} = \frac{\text{deeltal}}{\text{deler}}$$

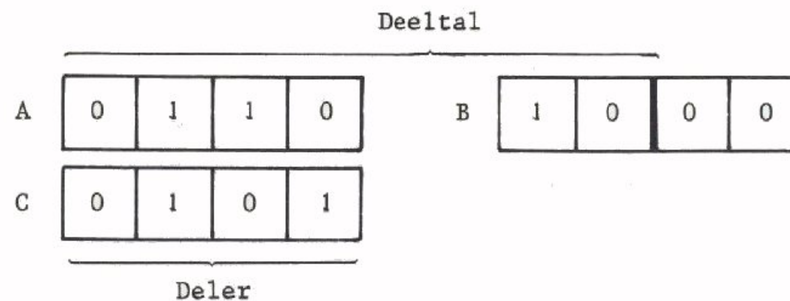
Stap 1. We plaatsen het deeltal in registers A en B, en de deler in register C



Stap 2. Als we de inhoud van register C willen aftrekken van de inhoud van register A, krijgen we een negatief resultaat want C > A. In dit geval laten we de inhoud van register A, 1 bit naar links schuiven en de inhoud van register B wordt 1 bit naar A geschoven.



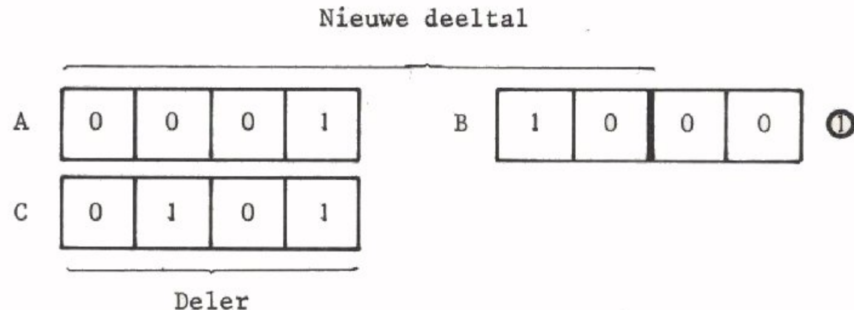
Stap 3. We krijgen weer dat de absolute waarde van C groter is dan van A en dus dat het resultaat van de aftrekking negatief zal zijn. We gaan daarom nog een keer schuiven zoals in stap 2



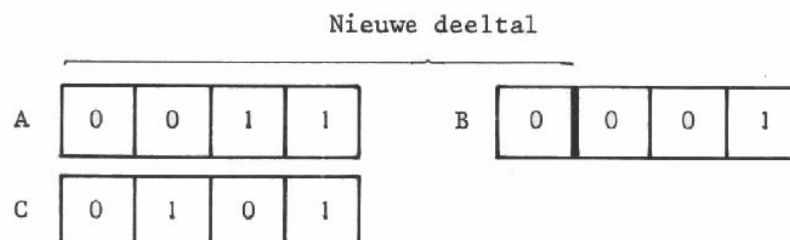
Stap 4. Nu levert A-C een positief getal op.

```
A    0 1 1 0
C    0 1 0 1
     -------
     0 0 0 1   = resultaat
```

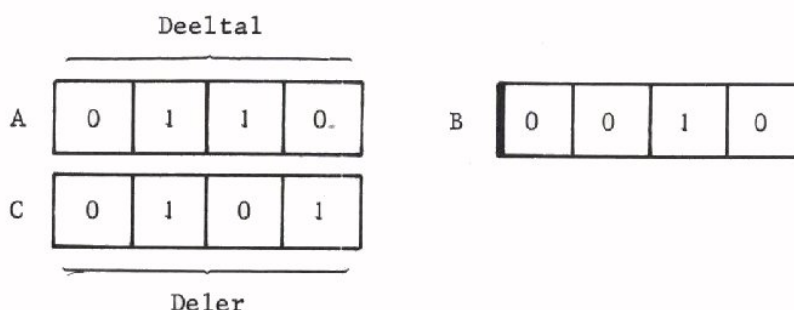
We zetten dit resultaat in register A; zijn vorige inhoud zijn we dus kwijt. Daarna schrijven we een "1" aan de achterkant van register B:



Tijdens het schuiven dat daarna plaatsvindt wordt de ① in het register B geschoven

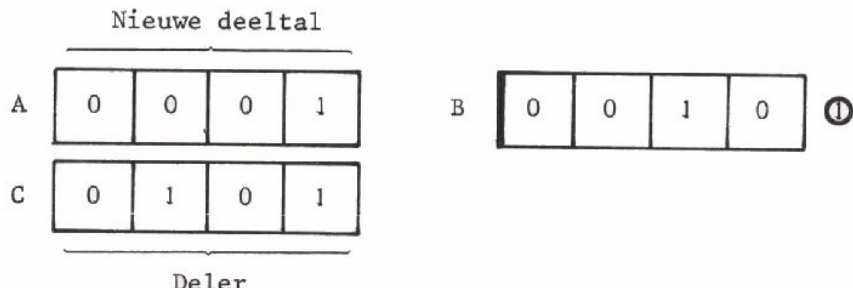


Stap 5. De absolute waarde van de inhoud van A is weer kleiner dan de absolute waarde van de inhoud van C. De aftrekking zal negatief zijn en we moeten opnieuw de inhoud van registers A en B een bit opschuiven.

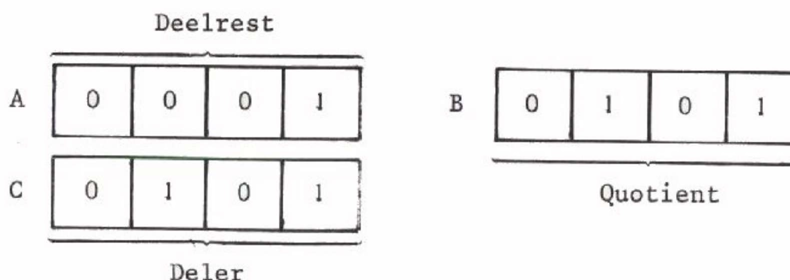


Stap 6. Het aftrekken van A-C zal nu een positief getal geven zodanig dat een "1" rechts van B geplaatst wordt en het resultaat van de aftrekking in A komt te staan:

A 0 1 1 0
C 0 1 0 1
0 0 0 1 ——→ resultaat



Daarna schuiven we de 1 in register B. Let erop dat register A nu niet meer geschoven wordt want register B bevat geen bit meer van het oorspronkelijke deeltal.



We zien nu dat register B het quotient bevat en register A de rest van de deling. Register C bevat nog de deler maar het oorspronkelijke deeltal zijn we kwijt.

OPMERKING

Het is U duidelijk geworden dat het vermenigvuldigen en het delen in rekenmachines niet rechtstreeks mogelijk is. We zullen zelf de schuif- en opteLopdrachten (resp. aftrekopdrachten) aan de machine moeten geven. Om deze reeksen van opdrachten voor het vermenigvuldigen of het delen mogelijk te maken, worden door de ontwerpers van rekenmachines meestal vaste programma's toegevoegd. Als twee getallen met elkaar vermenigvuldigd moeten worden, wordt het betreffende programma automatisch opgeroepen als de vermenigvuldigingstoets van de machine wordt ingedrukt. Hierop komen we later nog terug als de microprocessor zelf zal worden behandeld.

DE BCD-CODE

BCD is de afkorting van de Engelse uitdrukking Binary Coded Decimal. Dit betekent binair gecodeerd decimaal talstelsel. Deze code is een manier om de decimale cijfers 0 tot en met 9 met behulp van de binaire bits 0 en 1 weer te geven. Elk decimaal cijfer moet door vier bits (tetrade genoemd) worden weergegeven.

0 = 0 0 0 0	5 = 0 1 0 1
1 = 0 0 0 1	6 = 0 1 1 0
2 = 0 0 1 0	7 = 0 1 1 1
3 = 0 0 1 1	8 = 1 0 0 0
4 = 0 1 0 0	9 = 1 0 0 1

men mag geen enkele 0 weglaten!

De combinaties 1010, 1011, 1100, 1101, 1110 en 1111 komen in deze code NIET voor.

De *omzetting* van een decimaal getal naar deze code en omgekeerd zal geen problemen opleveren vanwege de eenvoud. Omzetting van binair naar BCD en omgekeerd kan via het decimale talstelsel geschieden.

Het optellen en aftrekken van BCD-getallen kan geschieden door de getallen eerst in binaire getallen om te zetten, daarna de optelling (resp. aftrekking te doen plaatsvinden en tenslotten de uitkomst weer in BCD-vorm om te zetten.

Moet de optelling of aftrekking direct geschieden, dan treden daarbij enkele extra moeilijkheden op wat betreft de carries en de borrows. Immers, de zes combinaties 1010 tot en met 1111 worden niet gebruikt.

HET OPTELLEN VAN TWEE BCD-GETALLEN

Aan de hand van een voorbeeld laten we nu twee manieren zien om twee BCD-getallen *bij elkaar op te tellen.*

<u>1e manier:</u>

```
. . . .     1 0 0 1    0 1 0 0      =  94(10)
            0 1 0 1    1 0 0 0      =  58(10)
-------------------------------- +
            1 1 1 0    1 1 0 0
            0 0 0 1 <- 1 0 1 0         10(10) overhevelen naar 2e tetrade
            ------- +  ------- -
            1 1 1 1    0 0 1 0
0 0 0 1 <-  1 0 1 0       |            10(10) overhevelen naar 3e tetrade
   |        ------- -     |
   |        0 1 0 1       |
   v                      v
0 0 0 1     0 1 0 1    0 0 1 0      =  152(10)
```

Zodra bij het optellen van twee tetrades de som groter is dan $1001_{(2)}$ = $9_{(10)}$, moet er $1010_{(2)} = 10_{(10)}$ overgeheveld worden naar de nevenstaande tetrade met groter gewicht. Het overschot blijft in de plaats van de som staan. Zo wordt in ons voorbeeld van de som $1100_{(2)}$ het getal $1010_{(2)}$ afgetrokken en blijft in deze tetrade $0010_{(2)}$ staan. Bij deze manier moet zowel worden opgeteld als worden afgetrokken. Hetzelfde kan men ook op de volgende manier bereiken, waarbij alleen maar van optellen sprake is.

2e manier:

	1 0 0 1	0 1 0 0		= $94_{(10)}$
	0 1 0 1	1 0 0 0	+	= $58_{(10)}$
	1 1 1 0	1 1 0 0		
		0 1 1 0	+	$0110_{(2)} = 6_{(10)}$ erbij optellen
	0 0 0 1 ← (1)	0 0 1 0		carry naar 2e tetrade
	+			
	1 1 1 1			
	0 1 1 0		+	$0110_{(2)} = 6_{(10)}$ erbij optellen
0 0 0 1 ← (1)	0 1 0 1			carry naar 3e tetrade
0 0 0 1	0 1 0 1	0 0 1 0		= $152_{(10)}$

Zodra bij het optellen van twee tetrades de som groter is dan $1001_{(2)} = 9_{(10)}$, moet er $0110_{(2)} = 6_{(10)}$ bij deze som worden opgeteld om als het ware de niet te gebruiken $1010_{(2)}$ tot en met $1111_{(2)}$ over te slaan.
We krijgen daardoor enerzijds de carry (in dit geval INTERMEDIATE CARRY genoemd) voor de nevenstaande tetrade; anderzijds verkrijgen we de "rest" die achter moet blijven in de tetrade waar we mee bezig zijn.

HET AFTREKKEN VAN TWEE BCD-GETALLEN

Vervolgens de twee manieren om twee BCD-getallen A en B *van elkaar af te trekken*, weer behandeld aan de hand van een voorbeeld.

1e manier:

A =	0 1 1 1	0 1 1 1		
B =	0 0 1 0	1 0 0 0	−	
		kan niet:		we moeten eerst lenen

A =	0 1 1 (1) →	0 1 1 1		
		1 0 1 0	+	$10_{(10)} = 1010_{(2)}$ lenen
	0 1 1 0	1 0 0 0 1		
B =	0 0 1 0	1 0 0 0	−	
	0 1 0 0	1 0 0 1		

Zodra een tetrade van een kleinere tetrade afgetrokken moet worden, moet er van de nevenstaande tetrade $0001_{(2)}$ geleend worden. Dit getal is voor de betreffende tetrade 10x zoveel waard, dat wil zeggen $10_{(10)} = 1010_{(2)}$. Pas na dit lenen kan de aftrekking plaats vinden.

<u>2e manier:</u>

A = 0 1 1 (1) ↘		0 1 1 1		
	1 0 0 0 0			1 lenen
	0 1 1 0		−	$6_{(10)} = 0110_{(2)}$ aftrekken, zo-
	1 0 1 0 ⟶	1 0 1 0	+	dat $10_{(10)} = 1010_{(2)}$ over is als echte waarde van 1.
0 1 1 0		1 0 0 0 1		
B = 0 0 1 0		1 0 0 0	−	
0 1 0 0		1 0 0 1		

De geleende 1 noemt men de INTERMEDIATE BORROW.

OPMERKINGEN

Bij het direct optellen (resp. aftrekken) van twee BCD-getallen is er blijkbaar sprake van twee soorten carries (resp. borrows).

Binnen de BCD-tetrades kunnen gewone carries en borrows optreden.

Bij overgang van de ene tetrade naar de andere nevenliggende tetrade kunnen zogenaamde INTERMEDIATE carries en borrows optreden.

Het *nut* van de BCD-code is, dat het 't ieder vertrouwde decimale talstelsel onmiddelijk weergeeft in nullen en enen die door digitale computers verwerkt kunnen worden. Een in BCD-code gegeven getal is trouwens direct als het ware decimaal te lezen. Dit laatste is bijvoorbeeld met de binaire code niet het geval.

7

A D D E R S

INLEIDING.

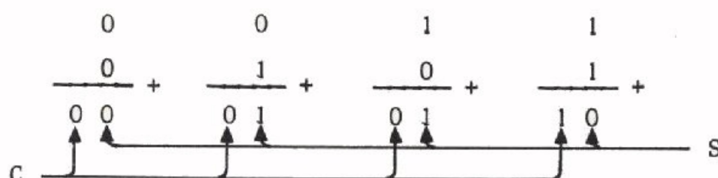
In het vorige hoofdstuk is duidelijk geworden, dat optellen één van de meest belangrijkste operaties in digitale rekenmachines is. Optellen vormt namelijk de basis voor andere operaties zoals aftrekken, vermenigvuldigen en delen.
In dit hoofdstuk gaan we aandacht besteden aan schakelingen die twee binaire getallen kunnen optellen. Dergelijke schakelingen worden ADDERS genoemd (dit is de engelse uitdrukking voor OPTELLERS).

HET OPTELLEN VAN TWEE BITS : DE HALF-ADDER

We hebben reeds gezien dat de regels voor het optellen van twee bits A en B zeer eenvoudig zijn. Ze kunnen in een cijfertabel worden samengevat. Dit is hieronder gedaan.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

In deze tabel is S de LSB van A plus B. Verder vertegenwoordigt C de MSB van A plus B (= de carry).



Gaan we twee bits m.b.v. een digitale schakeling (de half-adder) optellen, dan worden de REKENKUNDIGE bits 0 en 1, daarin vertegenwoordigd door de LOGISCHE waarden 0 (= niet waar) en 1 (= waar). De boven gegeven CIJFERTABEL is dus tevens de WAARHEIDSTABEL voor de half-adder.

We zien dat S = 1 als:

A is 0 EN B is 1

of

A is 1 EN B is 0

Dit kunnen we in de vorm van een logische vergelijking schrijven:

$$S = \bar{A}.B + A.\bar{B}$$

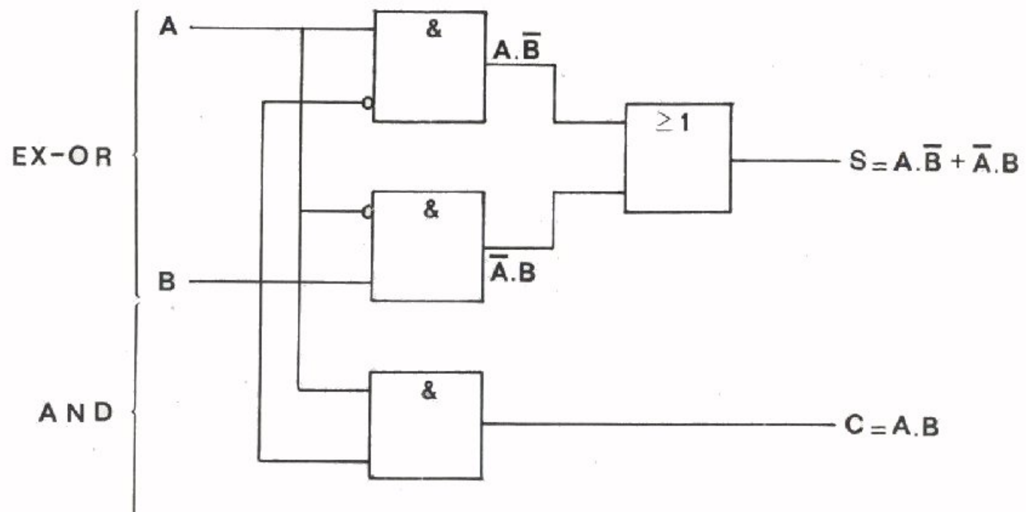
(Dit is in feite de exclusieve OR-functie).

Verder zien we dat C = 1 als A EN B 1 zijn. In alle andere gevallen geldt C = 0. Dus:

$$C = A.B$$

(Dit is in feite de AND-functie).

Met behulp van deze twee vergelijkingen kunnen we een ontwerpschema afleiden van een schakeling die twee bits kan optellen. Deze is op volgende pagina weergegeven.



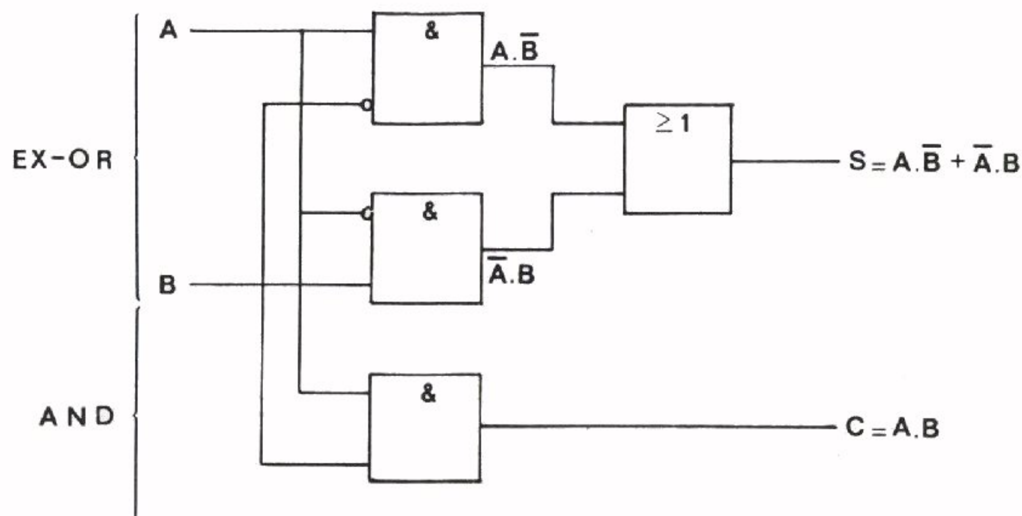
Dit schema bevat drie AND-poorten (waarvan twee met één negerende ingang) en een OR-poort. Een schakeling die in staat is twee bits op te tellen noemen we een HALF-ADDER (dit betekent "halve opteller"). Meestal gebruikt men in Nederland de engelse benaming en we zullen deze ook in onze lessen handhaven. Waarom deze schakeling HALF-adder genoemd wordt zal later in dit hoofdstuk duidelijk worden.

OPMERKING

In het begin van dit hoofdstuk hebben we de functie van de half-adder theoretisch beschreven. Hierbij hebben we de logische waarde 0 en 1 toegepast en een ontwerpschema van de half-adder opgesteld. Dit schema willen we met onze bouwstenen realiseren. We moeten dus een uitvoeringsschema afleiden. Hiervoor zijn de afspraken van toepassing die in hoofdstuk 1 zijn besproken. Bekijk ze nog eens als dit nodig is, alvorens verder te gaan.

UITVOERINGSSCHEMA VAN DE HALF-ADDER

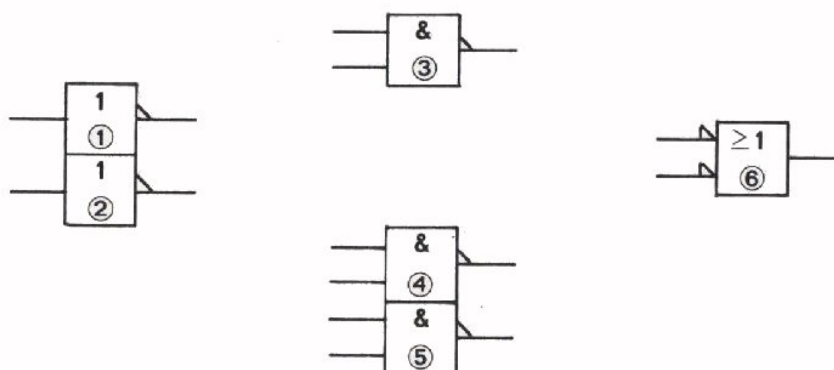
Bekijk eerst nog even het ontwerpschema.



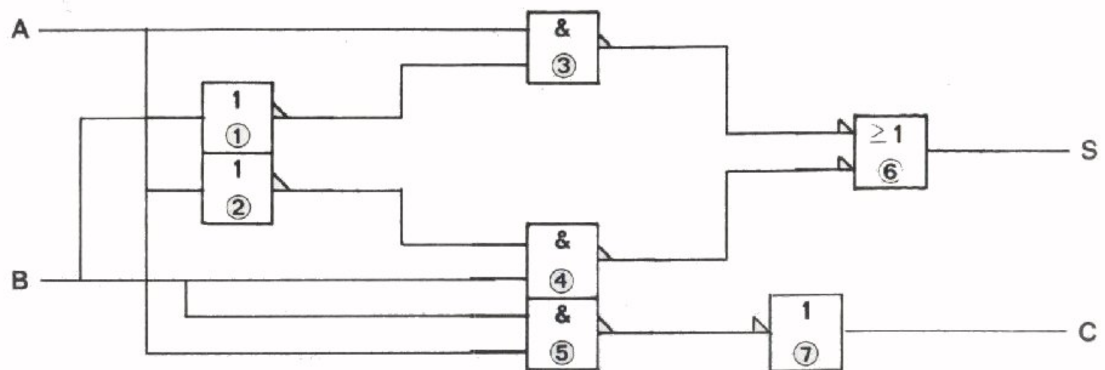
We beschikken in de onderdelendoos over inverterende AND-poorten en inverterende OR-poorten. We gaan uitsluitend gebruik maken van inverterende AND-poorten (de IC 7400). Deze poorten zijn op te vatten als:



We kiezen eerst weer de bouwstenen zonder doorverbindingen aan te brengen. De twee negaties realiseren we m.b.v. inverters.



Tenslotte brengen we de doorverbindingen aan, waarbij na poort ⑤ nog een inverter ingelast moet worden.



Voor de overzichtelijkheid hebben we aan elke inverterende AND-poort het symbool gegeven van de functie die hij verricht. Verder zien we dat aan de ingangen en aan de uitgang H = 1 en L = 0 geldt dat de ingangen en de uitgangen hoog actief zijn. Dit was in het ontwerpschema nog niet vastgelegd.

In de volgende opdracht gaan we een half-adder opbouwen met behulp van twee 7400-modulen en vervolgens zijn functietabel proefsgewijs samenstellen.

OPDRACHT 7.1 DE HALF-ADDER

- Bouw het laatste circuit op met behulp van twee 7400-modulen. Denk eraan dat als een inverterende AND als inverter gebruikt wordt, óf de twee ingangen kortgesloten moeten zijn, óf de niet gebruikte ingang met de + 5V doorverbonden moet zijn.

- Gebruik de 8-BIT-selector module om de niveaus van A en B te bepalen, en de 8-LED indicator om de niveaus van S en C uit te lezen.

- Vul volgende functietabel in:

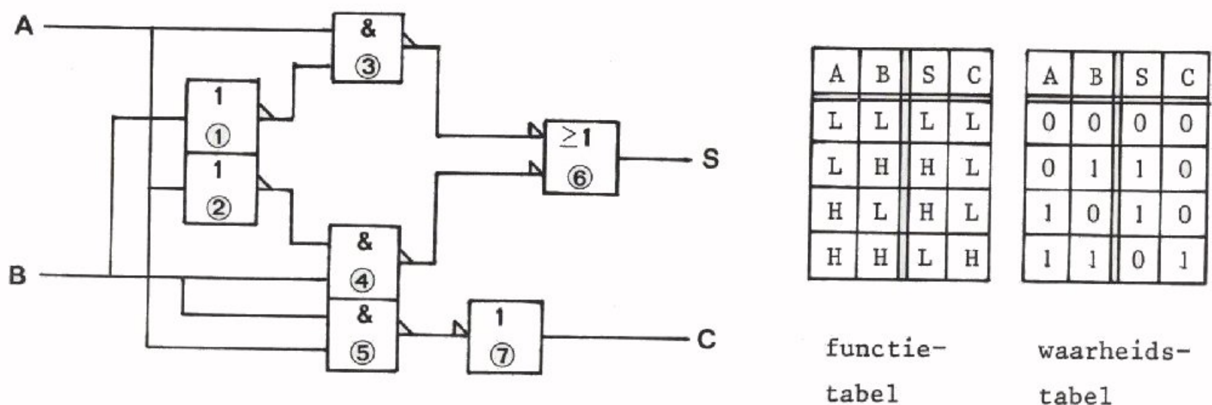
A	B	S	C
L	L		
L	H		
H	L		
H	H		

- We hebben afgesproken dat in geval van hoog actieve in- en uitgangen:
  de logische "0" wordt aangegeven door niveau L
  de logische "1" wordt aangegeven door niveau H

Vergelijk nu (met deze afspraak in gedachten) de verkregen functietabel met de waarheidstabel van de half-adder.

CONCLUSIE

Met onderstaande schakeling is het mogelijk twee bits op te tellen. De functietabel en de waarheidstabel zijn er naast getekend. Bedenk dat alle in- en uitgangen actief hoog zijn, zodat H = 1 en L = 0.
Verder komt de waarheidstabel weer overeen met de "cijfertabel", die het optellen van twee binaire cijfers weergeeft.



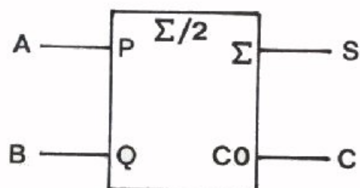
A	B	S	C
L	L	L	L
L	H	H	L
H	L	H	L
H	H	L	H

functie-
tabel

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

waarheids-
tabel

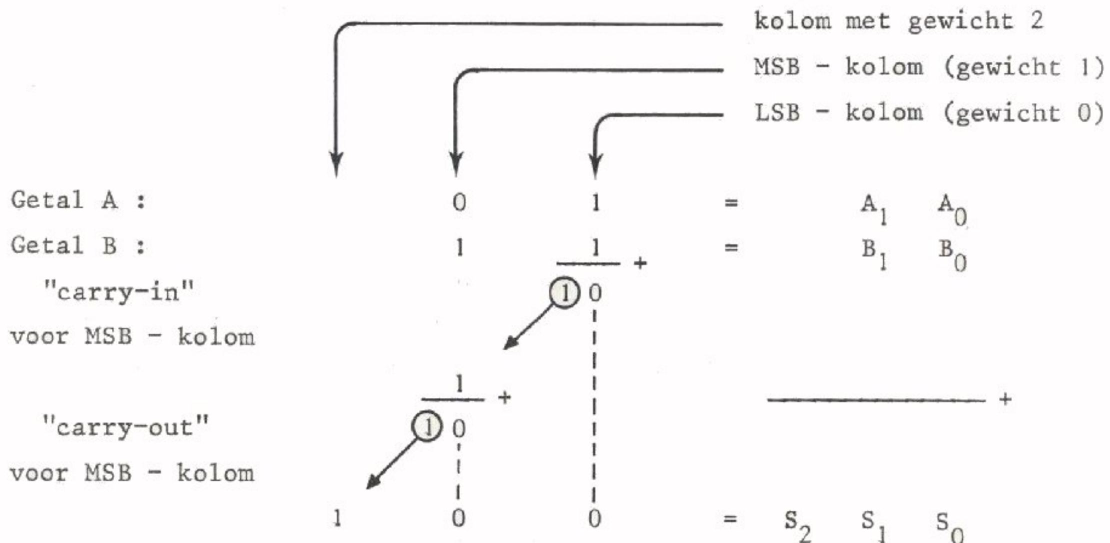
Een half-adder kan met behulp van het volgende symbool worden weergegeven.



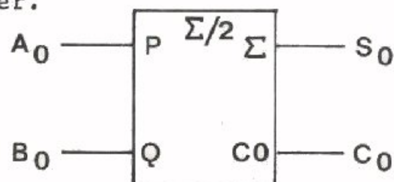
In dit symbool staat de griekse hoofdletter Σ (sigma) voor de optelfunctie; hij duidt aan waar de LSB van de som van de twee bits (P en Q in het symbool) beschikbaar komt. De carry is aangegeven door CO. Dit betekent Carry-out (= carry uit). Waarom we carry-out zeggen en niet alleen maar "Carry" zal U hierna uitgelegd worden. Tevens zal dan duidelijk worden waarom de schakeling half-adder genoemd wordt.

HET OPTELLEN VAN DRIE BITS - DE FULL-ADDER

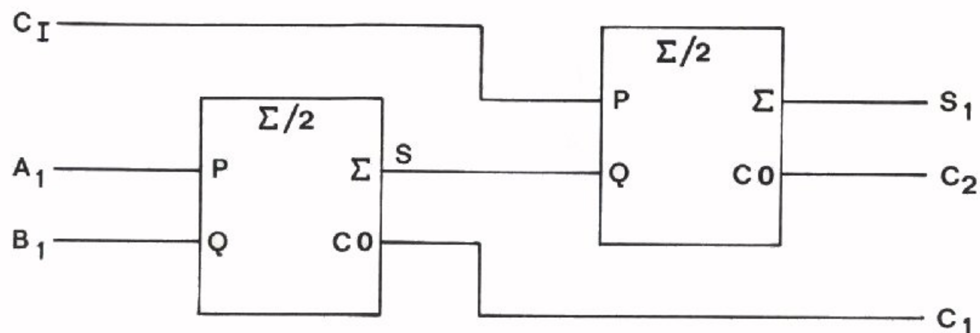
Bekijk de eenvoedige optelling van getal A en getal B hieronder.



Eerst worden de LSB's van de getallen A en B opgeteld, d.w.z. de TWEE bits 1 en 1 worden opgeteld. Er komt 10 uit en hiervan is de 1 een carry die naar de MSB - kolom toe gaat. Omdat hij de MSB - kolom binnenkomt, noemt men hem de CARRY-IN voor deze MSB - kolom. Vervolgens moeten de MSB's van de getallen A en B én de binnengekomen "carry-in" worden opgeteld; d.w.z., de DRIE bits 0, 1 en 1 moeten worden opgeteld.
Daar komt 10 uit, en hiervan is de 1 een carry die een plaats naar links gaat. Omdat hij de MSB - kolom uitgaat, noemt men hem de CARRY-OUT voor deze MSB - kolom. Met behulp van half-adders gaan we een schakeling opbouwen om twee getallen van elk twee bits op te tellen.
De schakeling moet eerst de TWEE LSB's van de getallen optellen. Dit kan met behulp van een half-adder.



Vervolgens moeten we de MSB's van de getallen optellen. Dan moeten we er echter rekening mee houden dat de eerste schakeling een carry-in heeft die ook toegevoegd moet worden. Het is nu dus nodig om DRIE bits bij elkaar op te kunnen tellen. Hiervoor kan men twee half-adders gebruiken. Dit is op de volgende bladzijde weergegeven.



Aan de schakeling worden toegevoerd:

- De MSB's A_1 en B_1 van de getallen A en B.
- De "carry-out" van de LSB's, die een "carry-in" C_I voor de MSB - kolom is.

We gaan vervolgens een waarheidstabel voor de schakeling opstellen om te zien wat hij doet.

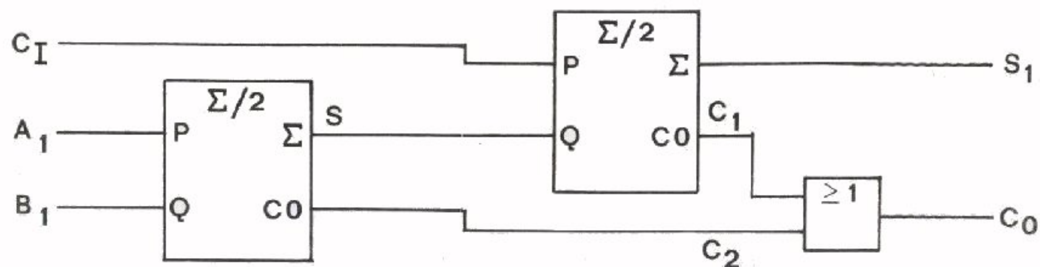
A_1	B_1	C_I	S	C_1	S_1	C_2
0	0	0	0	0	0	0
0	0	1	0	0	1	0
0	1	0	1	0	1	0
0	1	1	1	0	0	1
1	0	0	1	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	1	0	1	1	0

We zien:

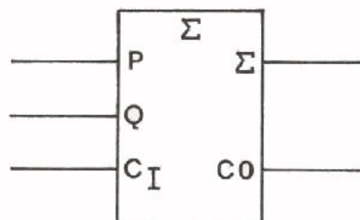
- Als er geen enkele 1 wordt toegevoerd, zijn alle uitgangen 0.
- Als er één 1 wordt toegevoerd, is alleen $S_1 = 1$.
- Als er twee keer 1 wordt toegevoerd, is óf $C_1 = 1$, óf $C_2 = 1$.
- Als er drie keer 1 wordt toegevoerd, is $S_1 = 1$ en $C_1 = 1$.

Dus: nooit zijn de carrys C_1 en C_2 tegelijkertijd 1. We kunnen daarom C_1 en C_2 aan een OR-poort toevoeren om zo één carry-uitgang te verkrijgen, de carry-out uitgang C_0.

Behalve 2 half-adders hebben we dus ook nog een OR-poort nodig om de drie bits A_1, B_1 en C_I op te tellen.



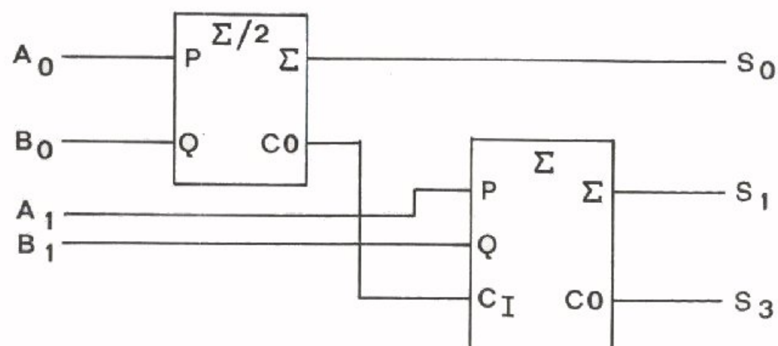
Deze uit twee half-adders en een OR-poort bestaande schakeling wordt een FULL-ADDER genoemd. Het is een schakeling die DRIE bits bij elkaar kan optellen. Omdat de schakeling die twee bits kan optellen (op de OR-poort na) de helft is van de schakeling die drie bits kan optellen, noemt men de eerste een HALF-adder en de tweede een FULL-adder. Het symbool van de FULL-adder is:



Hierin geldt:

- P en Q zijn de ingangen voor de op te tellen bits van een bepaalde kolom.
- C_I is de ingang voor de eventuele "carry-in" (die van rechts genoemde kolom binnenkomt).
- Σ is de uitgang die de som-bit voor genoemde kolom afgeeft.
- C_0 is de uitgang die de eventuele "carry-out" afgeeft (die naar links genoemde kolom uitgaat).

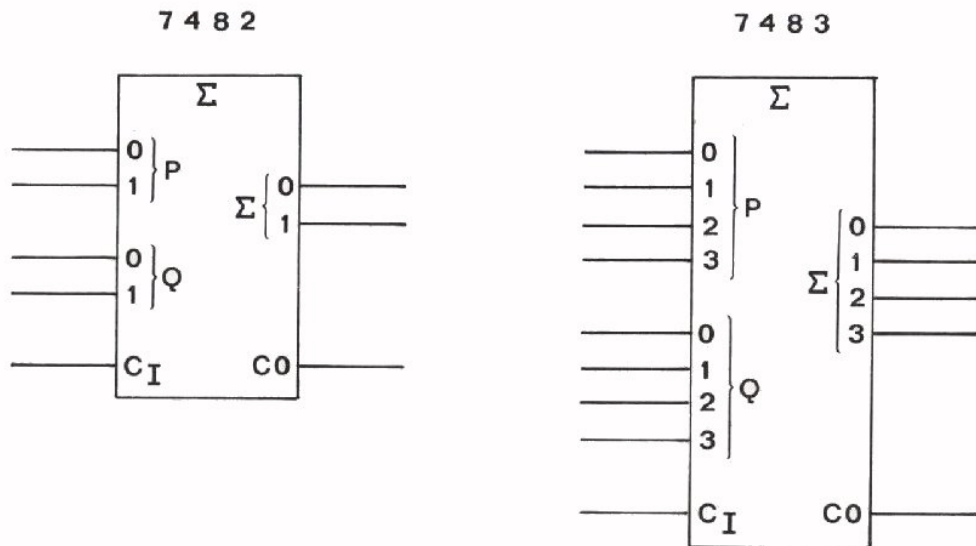
De schakeling voor het optellen van twee getallen van elk twee bits kan bestaan uit een combinatie van een half-adder en een full-adder zoals hieronder is getekend.



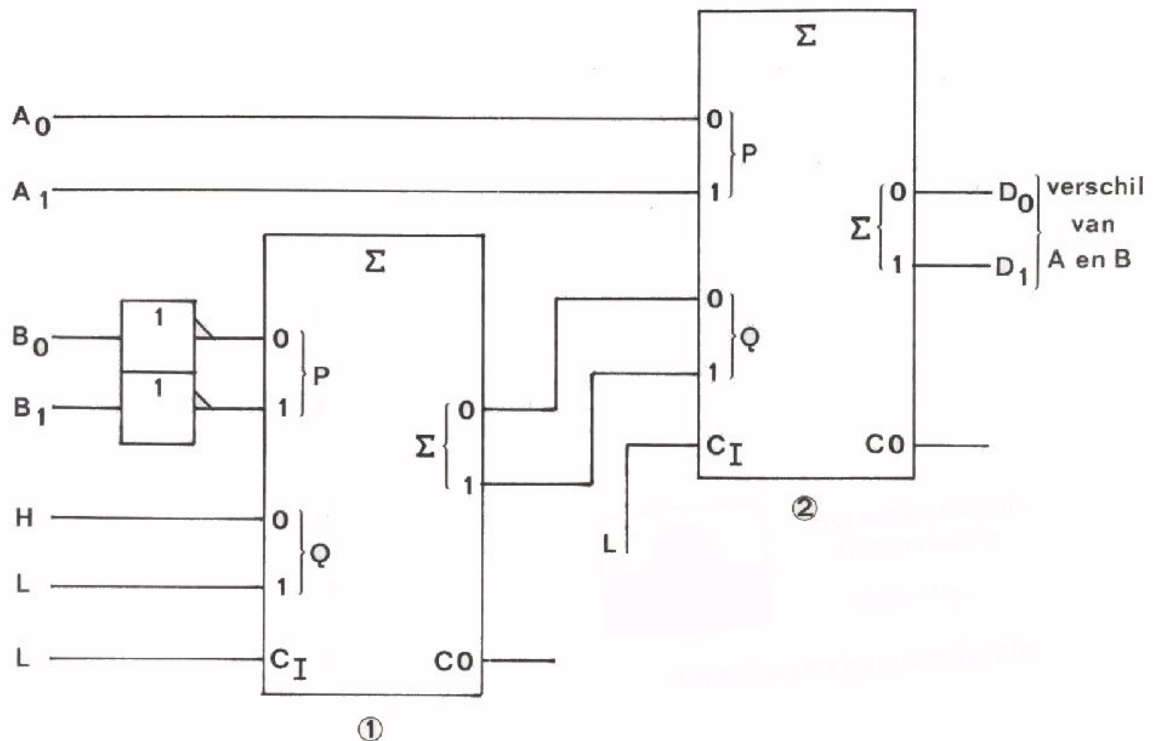
Moet men getallen van elk n bits optellen, dan zijn daarvoor nodig één half-adder (voor de LSB's) en n-1 full-adders (voor de overige bits). Immers: alleen bij de LSB's kan geen "carry-in" komen, bij elk van de volgende bit-paren is dat echter wel het geval.

ADDER-UITVOERINGEN

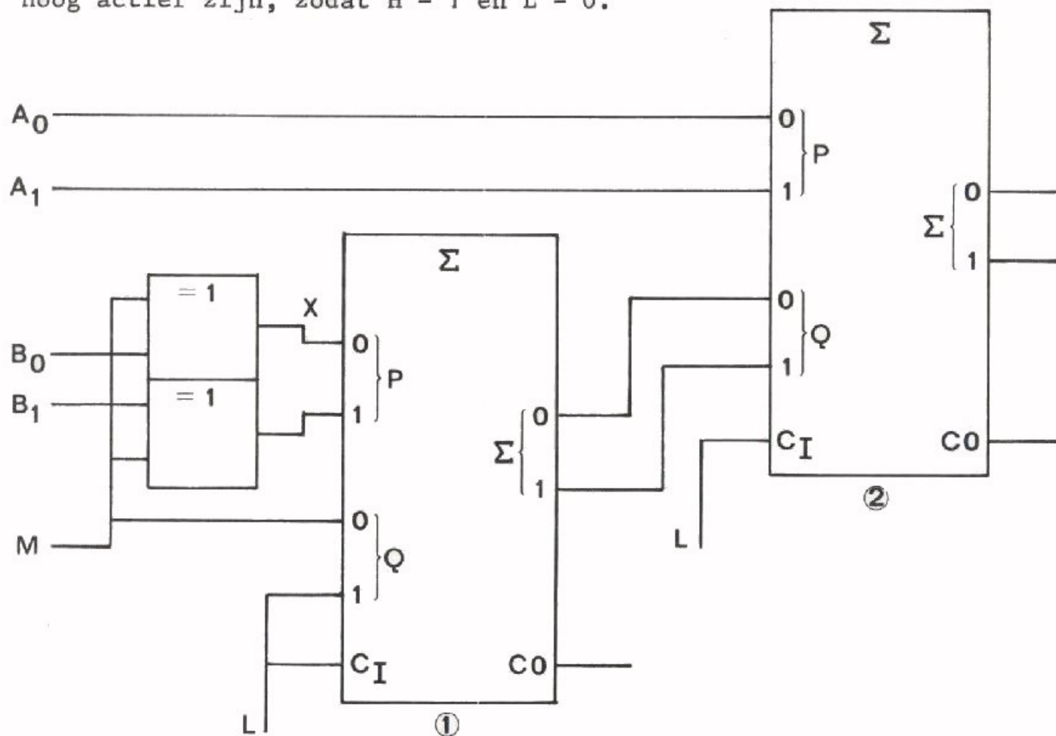
In de praktijk worden adder-combinaties kortweg ADDERS genoemd. Omdat adders veel voorkomende schakelingen zijn, bestaan ze ook in IC-vorm.
Hieronder heeft U twee voorbeelden.



De 7482 is een 2 bits- en de 7483 een 4 bits-adder. Beide hebben een C_I-ingang, zodat ze in serie geschakeld kunnen worden om grote getallen op te kunnen tellen. De getallen naast de accolades voor P,Q en Σ, geven het gewicht van de bits weer. U kunt opmerken dat zowel bij de ingangen als bij de uitgangen de bovenste aansluiting altijd voor de LSB is. Adders in IC-vorm kunnen gemakkelijk worden gecombineerd om andere rekenkundige bewerkingen uit te voeren. Hieronder ziet U bij voorbeeld op welke manier twee adders gebruikt kunnen worden om twee 2 bits-getallen van elkaar af te trekken. Ga dit na.
Bedenk dat overal de in- en uitgangen hoog actief zijn (óók achter de inverters, die als NOT-poorten dienst doen), zodat H = 1 en L = 0.



U heeft begrepen dat adder 1 zodanig is geschakeld, dat hij het 2-complement van getal B maakt. Dit 2-complement wordt dan in adder 2 bij getal A opgeteld. Het resultaat D is dan het verschil A - B. Het is ook mogelijk de bovenstaande schakeling zodanig te wijzigen, dat hij, naar wens, kan optellen of aftrekken. Dit kan bij voorbeeld met onderstaande schakeling bereikt worden. Bedenk weer dat overal de in- en uitgangen hoog actief zijn, zodat H = 1 en L = 0.



De twee inverters zijn vervangen door twee exclusieve OR-poorten. Deze exclusieve OR-poorten zijn aangesloten op het B-signaal, terwijl ze bovendien een gemeenschappelijke M-ingang hebben. De functietabel en de waarheidstabel van de bovenste EX-OR-poort zijn:

M	B_0	X
L	L	L
L	H	H
H	L	H
H	H	L

functietabel

OPTEL - TOESTAND (eerste twee rijen)

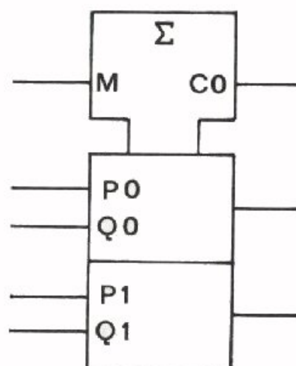
AFTREK - TOESTAND (laatste twee rijen)

M	B_0	X
0	0	0
0	1	1
1	0	1
1	1	0

waarheidstabel

- Als M laag is, volgt de uitgang X het niveau van B_0. Tegelijkertijd is ingang Q_0 van adder 1 laag. De uitgang van adder 1 levert dus het getal B aan adder 2. De schakeling levert aan de uitgang de som van A en B.
- Als M laag is, werken de exclusieve OR-poorten als inverterende poorten en wordt ingang Q_0 van adder 1 op hoog gezet. Hierdoor levert adder 1 het 2-complement van B aan adder 2. De schakeling levert dan aan de uitgang het verschil van A en B.

Voor de laatste schakeling heeft men het volgende symbool bedacht.



Contrôle-ingang M wordt de mode- of selectie-ingang genoemd. Als deze laag is, hebben we een optel-schakeling. Is M hoog, dan hebben we een aftrek-schakeling.

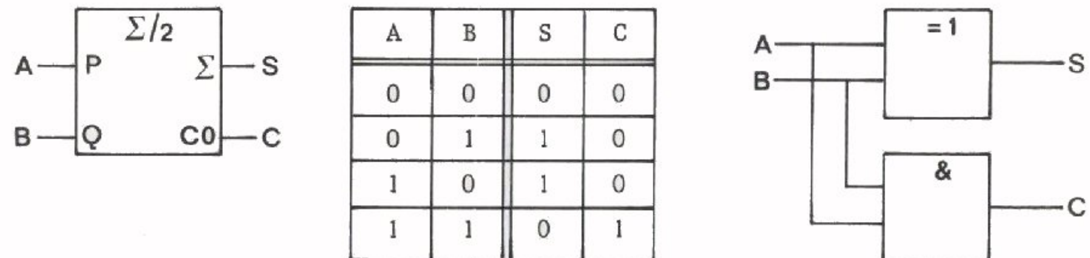
OPMERKING:

U heeft zojuist een schakeling bestudeerd die het mogelijk maakt twee verschillende rekenkundige operaties te verrichten. Er bestaan gecompliceerdere schakelingen, die nog meer operaties kunnen verrichten met grotere getallen. De verschillende operaties worden dan geselecteerd met behulp van 2 of meer selectie-ingangen. Een praktisch voorbeeld van een dergelijke schakeling is de IC 74181, de ALU.
Deze gaan we in de volgende les bestuderen.

SAMENVATTING

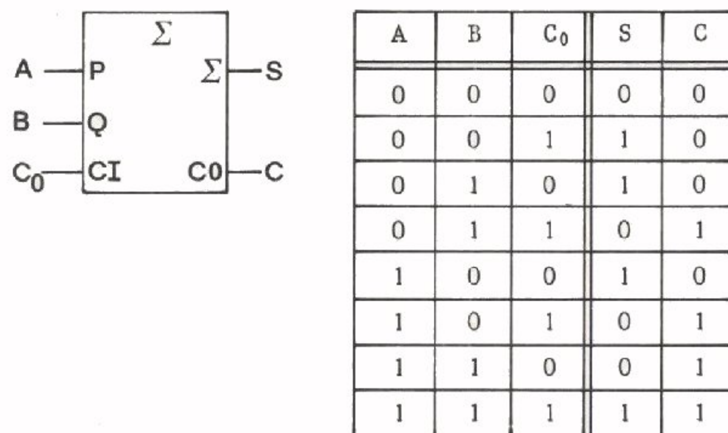
Een ADDER is een schakeling die twee binaire getallen kan optellen en vaak ook het ene getal van het andere kan aftrekken.

Optellen van 2 bits kan m.b.v. de combinatie van een EX-OR en een AND-poort (= een z.g. HALF-ADDER).

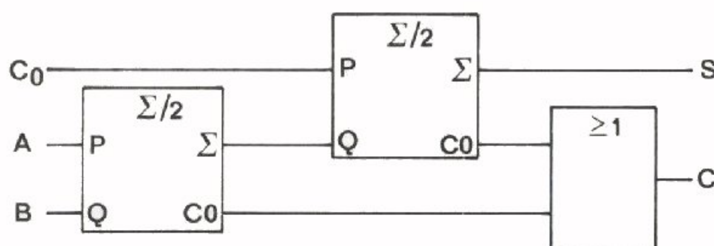


A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Optellen van 2 bits en carry-in kan m.b.v. de combinatie van twee half-adders en een OR-poort (= z.g. FULL-ADDER).



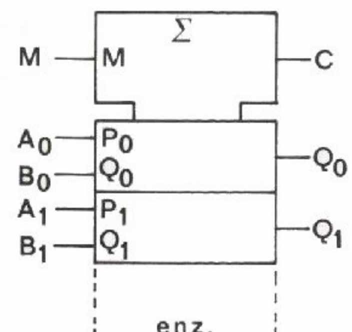
A	B	C_0	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Men kan *twee n-bits getallen optellen* m.b.v. 1 half-adder en n-1 full-adders.

Bij een adder die *zowel B bij A kan optelle* *als B van A kan aftrekken*, wordt inwendig d 2-complement methode toegepast voor de af-trekking:

- bij M = L wordt opgeteld
- bij M = H wordt afgetrokken



8

DE ALU

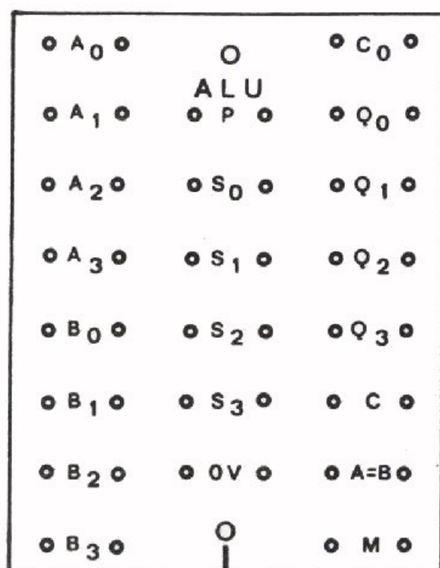
INLEIDING

De in het vorige hoofdstuk behandelde adders vormen de basis voor meer gecompliceerde schakelingen en een daarvan is de ALU. De ALU (afkorting van "Arithmetic and Logic Unit") is een zeer belangrijk onderdeel van computers, zowel van de grootste als van de eenvoudigste zakrekenmachine. Dit onderdeel is in staat een of twee binaire getallen rekenkundig dan wel logisch te bewerken.

De ALU heeft men ook in IC-uitvoering gerealiseerd. In dit hoofdstuk gaat U kennis maken met de ALU van het type 74181.

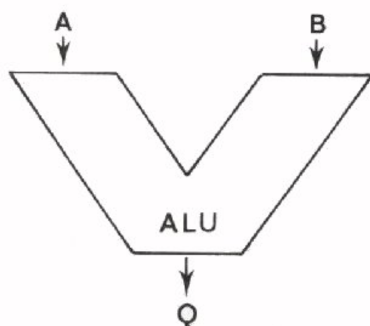
DE ALU-MODULE

Het woord ALU is een afkorting van de engelse uitdrukking "Arithmetic and Logic Unit". Het is een onderdeel dat berekeningen kan verrichten.



Hiernaast ziet U een afbeelding van de frontplaat van de ALU-module, die in de onderdelendoos aanwezig is. De module heeft 22 aansluitpunten, die we in de loop van dit hoofdstuk zullen bespreken. De P- en de 0 V-aansluitingen zijn verbonden met de contactpennen. Ze zijn respectievelijk de + 5 V en de 0 V van de voedingsspanning.

We zullen nu even aandacht besteden aan de A-, B- en Q-aansluitingen.



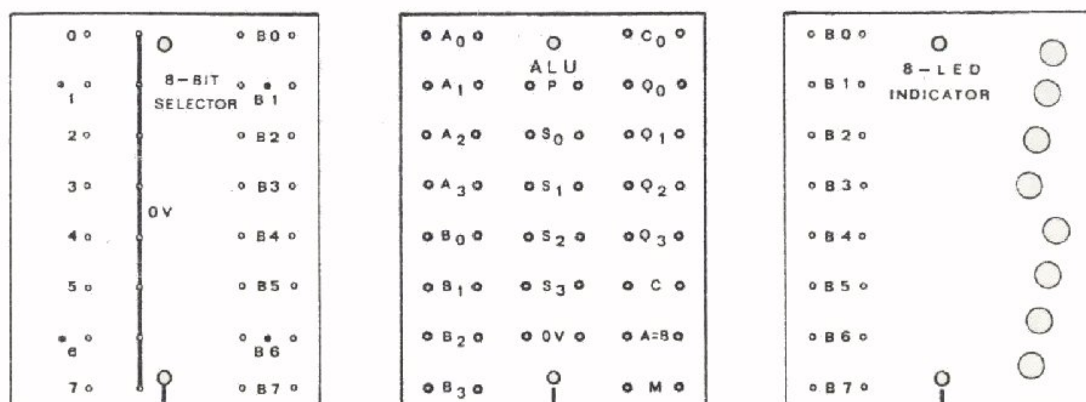
De ALU wordt soms getekend als een "pijlpunt", zoals hiernaast is weergegeven. Hij heeft twee ingangen A en B en één uitgang Q.

A en B zijn twee binaire getallen, die in het geval van de ALU type 74181, elk uit 4 bits zijn samengesteld. Getal A is dus $A_3A_2A_1A_0$ met A_3 als MSB en A_0 als LSB. Overeenkomstig geldt getal $B = B_3B_2B_1B_0$. Deze symbolen ziet U aan de linkerkant van de module. Uitgang Q is ook samengesteld uit 4 bits: $Q = Q_3Q_2Q_1Q_0$.Deze vier uitgangen bevinden zich aan de rechterkant van de module.

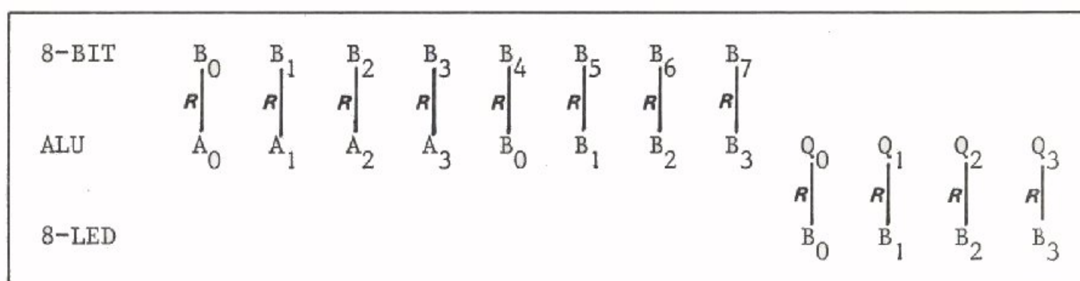
De betekenis van de andere aansluitingen zullen in de loop van dit hoofdstuk verklaard worden. In de volgende opdracht gaat U practische ervaring opdoen met de ALU.

OPDRACHT 8.1: DE ALU-MODULE

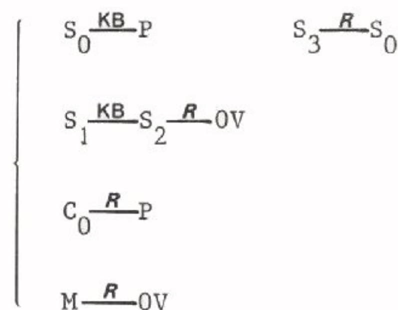
- Prik de 8-BIT selector, de ALU en de 8-LED indicator op het schakelpaneel zoals hieronder is weergegeven. Vergeet niet vooraf voldoende vierpoolverbindingssteunen en de voedingsstabilisator te plaatsen.



- Breng de volgende aansluitingen tot stand.



Verder op de ALU:



- Vul onderstaande tabel proefsgewijze in.

A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	Q_3	Q_2	Q_1	Q_0	
L	L	L	L	L	L	L	L					1
L	L	L	H	L	L	L	L					2
L	L	H	H	L	L	L	L					3
L	H	H	H	L	L	L	L					4
L	L	L	H	L	L	L	H					5
L	L	H	H	L	L	L	H					6
L	H	H	H	L	H	H	H					7
H	H	H	H	L	L	L	H					8
H	H	H	H	L	L	H	H					9

RESULTATEN

Regel 1 geeft de startconditie aan.

Regels 2 t/m 4: Het uitgangssignaal volgt signaal A.

Voor elke regel geldt:

$Q_3Q_2Q_1Q_0 = A_3A_2A_1A_0$ of Q = A.

Regels 5 t/m 7: Het uitgangssignaal is niet meer gelijk aan signaal A en wordt dus beïnvloed door signaal B.

Laat ons eerst regel 5 bestuderen. Daar hebben we $A_3A_2A_1A_0$ = L L L H en $B_3B_2B_1B_0$ = L L L H. De A- en B-ingangen en de Q-uitgangen van de ALU zijn hoog actief. In het geval van hoog actieve in- en uitgangen is H = 1 en L = 0. De getallen die door A en B zijn weergegeven worden dus:

$A = A_3A_2A_1A_0 = 0\ 0\ 0\ 1$

$B = B_3B_2B_1B_0 = 0\ 0\ 0\ 1$

——— PLUS

De som wordt: $0\ 0\ 1\ 0 = Q_3Q_2Q_1Q_0$

Deze wordt door de 8-LED indicator weergegeven als $Q_3Q_2Q_1Q_0$ = L L H L = 0 0 1 0. Ga na of dit ook klopt voor de regels 6 en 7.

Dus: $Q_3Q_2Q_1Q_0 = A_3A_2A_1A_0$ plus $B_3B_2B_1B_0$

of Q = A plus B.

OPMERKING

Om verwarring te vermijden geven we voortaan de logische "OR-functie" weer met "+", en de rekenkundige "som" met "plus".

Regels 8 en 9: De ALU geeft nu de som van A en B gedeeltelijk weer:

A = 1 1 1 1

B = 0 0 1 1

——————— PLUS

Som = carry 1 plus 0 0 1 0

De carry met gewicht 4 gaat bij het uitvoeren van deze opdracht verloren omdat er slechts 4 bits Q_0, Q_1, Q_2 en Q_3 kunnen worden afgegeven.

CONCLUSIE

In de opdracht hebben we gezien hoe de ALU twee 4-bits binaire getallen kan optellen. Een eventuele carry (5de bit) gaat echter verloren als men alleen de uitgangen Q_0, Q_1, Q_2 en Q_3 gebruikt. In feite geeft de ALU met een andere uitgang toch informatie betreffende de carry. Dit zult U in de volgende opdracht ontdekken.

OPDRACHT 8.2: OPTELLEN MET DE ALU

- In het circuit van de vorige opdracht verbindt U de aansluiting C van de ALU met B_5 van de 8-LED indicator met behulp van een kortsluitsteun.

- Vul volgende tabel in. Let er op dat in deze tabel en ook in de nog volgende tabellen van dit hoofdstuk, wordt gekozen voor de logische notatie om de getallen rechtstreeks te kunnen vergelijken.
  Daarbij geldt voor de hoog actieve A- en B-ingangen en Q-uitgangen dat L = 0 en H = 1. Voor de laag actieve C-uitgang geldt echter H = 0 en L = 1.

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q_0	Q_1	Q_2	Q_3	C
0	0	0	0	0	0	0	0					
1	0	0	0	0	0	0	0					
1	1	0	0	0	0	0	0					
1	1	1	0	0	0	0	0					
1	1	1	1	0	0	0	0					
1	1	1	1	1	0	0	0					
1	1	1	1	1	1	0	0					

Voor de laatste 2 regels geldt C = L = 1, zodat:

$A_3A_2A_1A_0$ plus $B_3B_2B_1B_0$

1 1 1 1 plus 0 0 0 1 = 0 0 0 0 en C = 1

1 1 1 1 plus 0 0 1 1 = 0 0 1 0 en C = 1

Voor de vorige regels gold C = H = 0.

Probeer het nog eens met willekeurige getallen A = $A_3A_2A_1A_0$ en B = $B_3B_2B_1B_0$.

CONCLUSIES

- Uit de tabel volgt dat aan de uitgangen Q_0, Q_1, Q_2 en Q_3 de som verschijnt van de getallen A_0 t/m A_3 en B_0 t/m B_3 op de eventuele carry na.

- Als het resultaat van de optelling groter is dan 1 1 1 1, dan treedt er een carry 1 op aan uitgang C. Deze uitgang is laag actief; d.w.z., C = 1 als C = L.

  Is het resultaat gelijk aan of kleiner dan 1 1 1 1, dan treedt er géén carry op. Dan geldt C = H = 0.

  Uitgang C wordt de "carry out"-uitgang genoemd. C is namelijk de carry-out voor de bits met gewicht 3.

- Let er op dat de ingangen S en ingang M in geval van rekenkundig optellen als volgt zijn aangesloten:

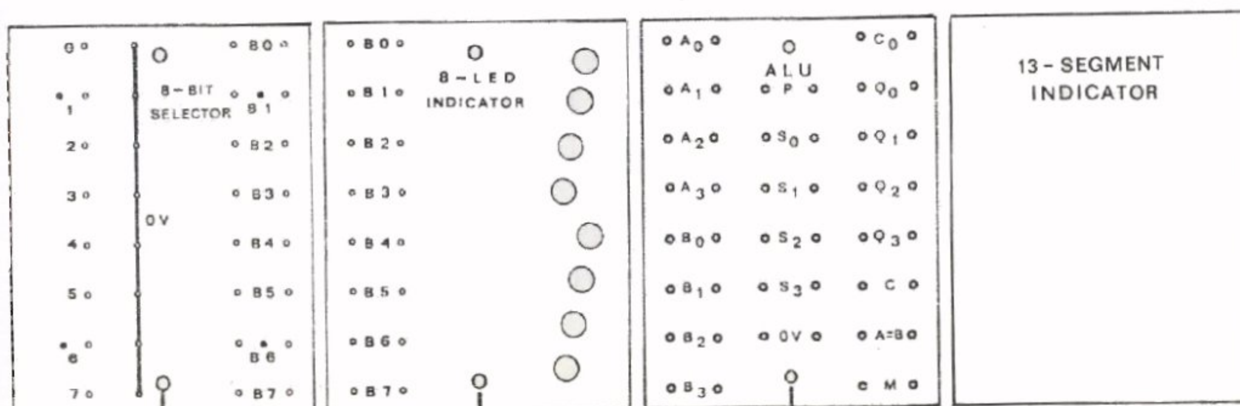
  S_0 = H
  S_1 = L
  S_2 = L
  S_3 = H
  M = L

  We komen hier nog op terug.

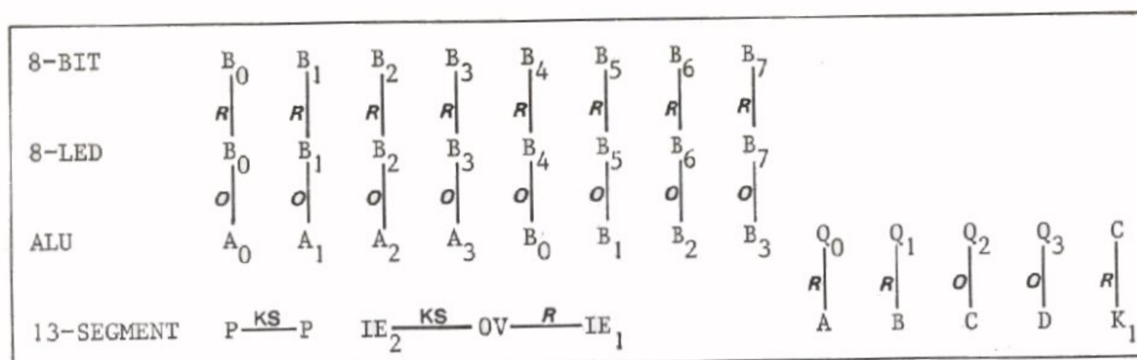
OPDRACHT 8.3: DE INGANG C_0 van de ALU

- Eerst gaan we het circuit iets wijzigen. We gaan de 13-segment indicator gebruiken voor het hexadecimaal uitlezen van resultaat Q. De 8 LED-indicator reserveren we voor het binair uitlezen van de ingangssignalen A en B.

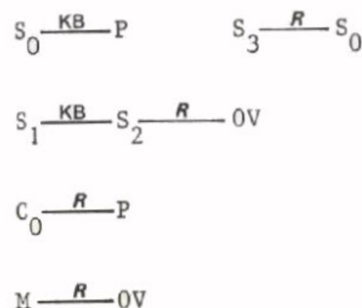
- Prik de modulen als volgt op schakelpaneel:



- De aansluitingen zijn nu:



Verder op de ALU:



- Vul in de volgende tabel hexadecimaal de waarden van Q in en vergelijk dit resultaat met de uitkomst van opdracht 8.2.
  Vergeet bij het invullen van de Q-waarden niet een eventueel aanwezige punt ook te noteren!

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	
1	1	1	0	0	0	0	0	
1	1	1	1	0	0	0	0	
1	1	1	1	1	0	0	0	
1	1	1	1	1	1	0	0	

De 13-segment indicator geeft nu door middel van een punt aan de linkerkant van het getal aan dat er een carry-out aanwezig is. Als nl. K_1 = L, dan licht de linker punt op.

- Maak C_0 los van P (+ 5 V).
- Verbind nu C_0 met 0 V.
- Vul de volgende tabel in en vergelijk deze met de vorige.

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	
1	1	1	0	0	0	0	0	
1	1	1	1	0	0	0	0	
1	1	1	1	1	0	0	0	
1	1	1	1	1	1	0	0	

Het verschil tussen de twee tabellen bestaat hierin, dat het resultaat in de laatste tabel elke keer 1 hoger is dan in de eerste tabel.

CONCLUSIE

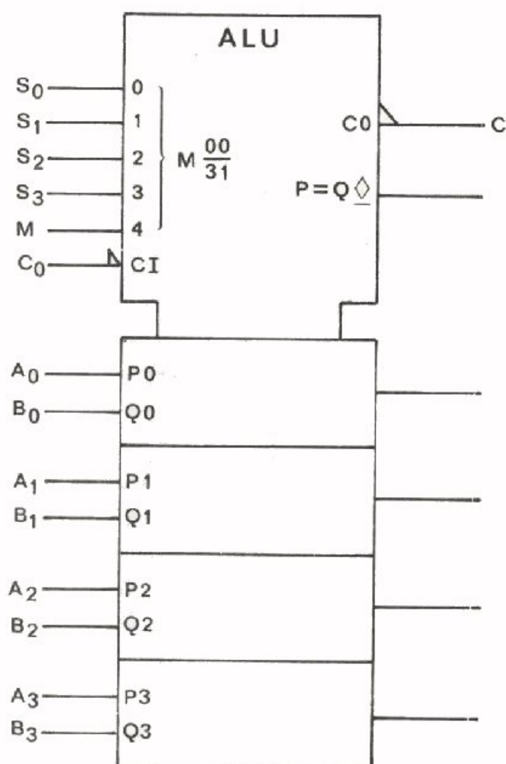
C_0 de ingang voor een carry-in. De carry-in is de bit, die we bij een eventuele vorige optelling hadden onthouden: als C_0 = L = 1, dan wordt er bij het totale resultaat 0 0 0 1 opgeteld.
De schakeling die we tijdens de vorige opdrachten bekeken hebben is dus een 4-bits opteller met C-in en C-out faciliteiten.
Input C_0 is evenals output C laag actief.

OPMERKING

De K_1 en K_2 ingangen van de 13-segment indicator zijn, zoals eerder vermeld, laag actief.

HET SYMBOOL VAN DE ALU

Tijdens de vorige opdrachten hebben we niet alleen gezien hoe men met de ALU 74181 kan optellen, maar is ook de betekenis van de aansluitingen C_0 en C duidelijk geworden.



Hiernaast hebben we het symbool van de ALU weergegeven. In het onderste blok vindt U alle ingangen en uitgangen van de ALU, die we reeds hebben bekeken. In het bovenste blok zijn links de controle- of functieselectie-ingangen aangegeven. Deze zijn buiten het symbool door S_0 t/m S_3 en door M weergegeven. Zij dienen voor de functie-selectie. Hierop komen we later nog terug.

In elk van de vorige experimenten hadden we: S_0 = H; S_1 = L; S_2 = L; S_3 = H en M = 0. Dit zijn de voorwaarden om de ALU als opteller dienst te laten doen. In de twee volgende opdrachten gaat U met de ALU een 4-bits getal van een ander 4-bits getal aftrekken.

AFTREKKEN MET BEHULP VAN DE ALU

In de eerste opdracht lieten we de ALU twee getallen optellen; daartoe waren S_0 = H, S_1 = L; S_2 = L en S_3 = H. Denk er aan dat deze optelling gebeurde in een 4-bits systeem.

In de laatste regel verricht de schakeling:

	A = 0 0 1 1 1	=	7_{10}
B = 0 1 1 1 1	- B = 1 0 0 0 1	=	-15_{10}
	___________		PLUS
resultaat =	1 1 0 0 0	=	-8_{10}

géén ← 1

er treedt géén carry-out op, zodat C = H = 0

De betekenis van het carry-out signaal is blijkbaar:

- als C = L = 1 (punt op 13-segment aanwezig),

  dan is het resultaat positief.

- als C = H = 0 (géén punt op 13-segment),

  dan is het resultaat negatief.

RECHTSTREEKS AFTREKKEN MET BEHULP VAN DE ALU

In de vorige opdracht hadden we met behulp van de ALU een getal B van getal A afgetrokken door -B bij A op te tellen. We moesten daarbij zelf zorgen voor het maken van het 2-complement van het getal B. De ALU had daarbij de optel-instelling:

$$S_0 = H,\ S_1 = L,\ S_2 = L,\ S_3 = H \text{ en } M = L.$$

Het aftrekken kan echter ook rechtstreeks door de ALU worden gedaan, want de daartoe benodigde schakeling is ingebouwd. Om deze schakeling te activeren moeten de functie-selectie ingangen als volgt staan:

$$S_0 = L,\ S_1 = H,\ S_2 = H,\ S_3 = L \text{ en } M = L$$

We gaan U dit in de volgende opdracht laten doen.

OPDRACHT 8.5: HET RECHTSTREEKS AFTREKKEN VAN EEN GETAL MET BEHULP VAN DE ALU

- Verbind de B_0 t/m B_3 ingangen van de ALU rechtstreeks met de 8-LED indicator. De 7400 module kan van het schakelpaneel worden verwijderd.
- Zet S_0 = L, S_1 = H, S_2 = H en S_3 = L.
- Check dat nog steeds C_0 = L en M = L.
- Vul opnieuw proefsgewijs de volgende tabel in en vergelijk ze met die van opdracht 8.4. Vergeet niet eventuele "punten" ook te noteren.

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
1	0	0	0	1	0	0	0	
1	1	1	0	1	0	0	0	
1	1	1	0	1	1	1	1	

Breek de schakeling nog NIET af.

CONCLUSIE

De ALU 74181 kan rechtstreeks een getal van een ander getal aftrekken indien de functie-selectie ingangen op de correcte wijze zijn ingesteld. Tot nu tot hebben we twee mogelijkheden gezien.

	S_0	S_1	S_2	S_3	M
A plus B	H	L	L	H	L
A minus B	L	H	H	L	L

DE "A=B" UITGANG VAN DE ALU

Een uitgang van de ALU-module die we nog niet hebben bekeken is de "A=B" uitgang. Binnen het symbolenkader van de ALU wordt deze uitgang met P = Q aangeduid (zie blz. 8.10). De uitgang is hoog actief, zodat aldaar geldt L = 0 en H = 1. In de volgende opdracht gaan we met deze uitgang ervaring opdoen.

OPDRACHT 8.6: DE "A=B" UITGANG

- We gebruiken het circuit van het vorige experiment dat nog steeds op Uw schakelpaneel aanwezig moet zijn. Er zijn twee wijzigingen:
  - C_0 moet nu hoog zijn (1-complement notatie)
  - de "A=B" uitgang van de ALU wordt met K_2 van de 13-segment module verbonden.
- Vul proefsgewijze kolom Q in van onderstaande tabel.

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q	C	A=B
0	0	0	0	0	0	0	0			
1	0	0	0	0	0	0	0			
1	0	0	0	1	0	0	0			
1	1	0	0	1	0	0	0			
1	1	0	0	1	1	0	0			
1	1	0	0	1	1	1	0			

- Vul vervolgens de kolommen C en "A=B" in met behulp van de resultaten die in kolom Q staan. Denk er aan dat, zodra een van de LED's K_1 en K_2 brandt, dit betekent dat de overeenkomstige variabele L is; is een LED uit, dan is zijn variabele H.
- Welke informatie kunt U uit de standen van C en "A=B" halen?
- Breek de schakeling nog NIET af.

CONCLUSIE

U kunt de ALU ook de opdracht geven twee getallen A en B te vergelijken. Hiervoor moet de ALU in de aftrek-toestand gezet worden (S_0 = L, S_1 = H, S_2 = H en S_3 = L) en de carry-in (C_0) moet H zijn.

De ALU geeft dan de uitkomst weer met behulp van de laag actieve uitgang C en de hoog actieve uitgang A = B.

C	A=B	C	A=B	Vergelijking
L	L	1	0	A > B
L	H	1	1	komt niet voor
H	L	0	0	A < B
H	H	0	1	A = B

Dus:

A > B   als "A=B" = L = 0 en er een carry-out optreedt (C = L = 1)

A < B   als "A=B" = L = 0 en er géén carry-out optreedt (C = H = 0)

A = B   als "A=B" = H = 1 en er géén carry-out optreedt (C = H = 0)

OPMERKING

In de tabel hiervoor ziet U dat voor de aftrekking (A minus B) en voor de vergelijking van A en B, de toestanden van de functie-selectie uitgangen gelijk zijn. Het enige verschil is de toestand van de carry-in ingang C_0. Dit kan op de volgende manier worden verklaard:

Om getal B van getal A af te trekken moeten we van getal B het 2-complement maken. Dit hebben we gerealiseerd door "intern" in de ALU het 1-complement te maken en daar "extern" via de carry-in ingang 1 bij op te tellen. D.w.z., dan is CI = L = 1. Zorgen we echter voor CI = H = 0, dan tellen we er niet 1 bij op, zodat we het 1-complement van getal B van getal A aftrekken. We verkrijgen dan dus:

A minus B minus 1

Het is deze handeling die gebruikt wordt om de getallen te vergelijken via de uitgangen C en "A=B".

Eerst bekijken we een voorbeeld waarbij A > B:

```
A    1 0 0 0                      1 0 0 0
B    0 0 1 0   1-complement       1 1 0 1
               ----------->     ---------  PLUS
                  van B         1 0 1 0 1
                                ↑
                                └─────────── carry-out
```

Nu geldt dus C = L = 1, terwijl uitgang "A=B" = L = 0 wordt omdat A = B (dit is zo intern in de ALU gerealiseerd).

Daarna bekijken we een voorbeeld waarbij A < B:

```
A    0 1 1 1                      0 1 1 1
B    1 0 0 0   1-complement       0 1 1 1
               ----------->     ---------  PLUS
                  van B           1 1 1 0
                                ↑
                                └─────────── géén carry-out
```

Nu geldt C = H = 0, terwijl uitgang "A=B" gelijk blijft aan L = 0.

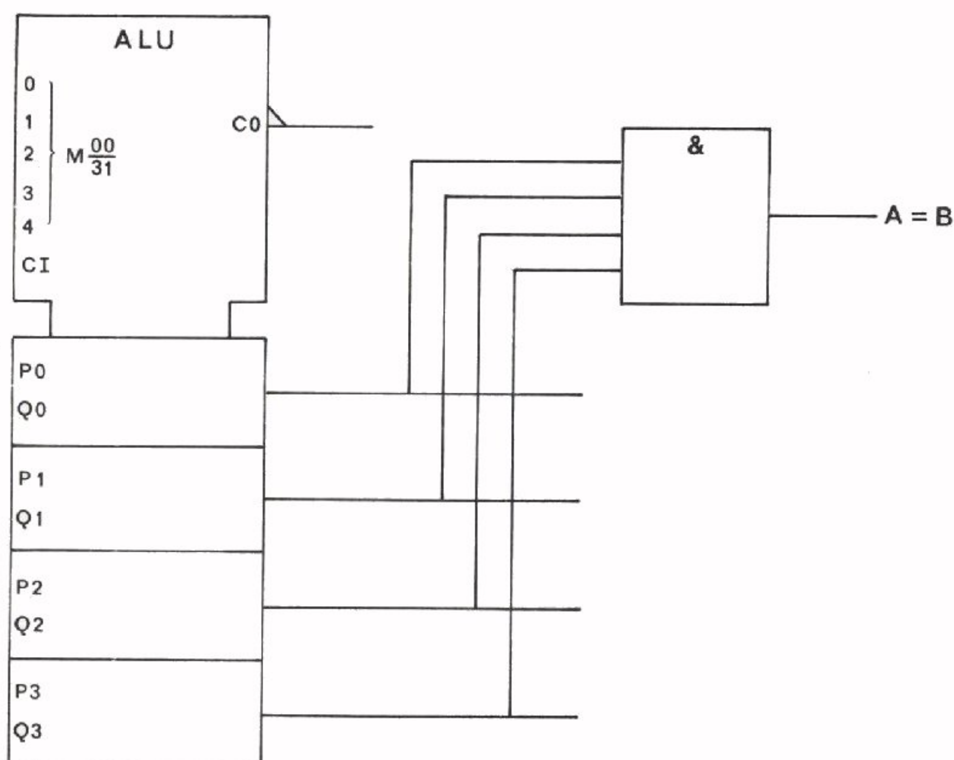
Tot slot bekijken we een voorbeeld waarbij A = B:

```
A    0 1 0 1                      0 1 0 1

B    0 1 0 1    1-complement      1 0 1 0
                ------------>     -------  PLUS
                   van B          1 1 1 1
                                ^
                                |
                                +--------- géén carry-out
```

Er geldt C = H = 0 en de som is altijd 1111. Ga dit nog even na.

Omdat de som nu altijd 1111 is, kan de "A=B"-uitgang in de ALU als hieronder worden verkregen:



OPMERKING

Het teken $\underline{\Diamond}$ aan de P = Q uitgang van het ALU-symbool (zie b.v. pagina 8.10) duidt op een z.g. "open collector" -uitgang. De collector van de uitgangstransistor moet in dit geval via een externe weerstand aan de + 5V worden aangesloten. Deze weerstand is in onze ALU-module inwendig reeds aangebracht.

VOORLOPIGE SAMENVATTING

Afhankelijk van de standen van de functieselectie-ingangen en van C_0 kunnen we volgens het voorgaande aan de ALU de opdracht geven om de volgende handelingen te doen:

S_0	S_1	S_2	S_3	C_0	
H	L	L	H	H	A plus B
L	H	H	L	L	A minus B
L	H	H	L	H	Vergelijk A met B

Door aan de functie selectie-ingangen S_0 t/m S_3 andere combinaties van L en H toe te voeren, zijn nog meer **bewerkingen van de getallen A en B** mogelijk. Hiervan zullen we er nog **twee proefsgewijs bekijken voordat U** de volledige tabel gaat bestuderen.

OPDRACHT 8.7: DE SHIFT OVER EEN BIT VAN GETAL A

- Zet S_0 = L, S_1 = L, S_2 = H, S_3 = H en C_0 = H op de ALU.
- Controleer dat M op de ALU laag blijft.
- Verwijder de verbinding tussen uitgang "A=B" van de ALU en de 13-segment indicator.
- Vul de kolom Q van de volgende tabel proefsgewijze in.

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0	0	0	0					
1	0	0	0	0	0	0	0					
1	1	0	0	0	0	0	0					
0	1	0	0	0	0	0	0					
0	1	1	0	0	0	0	0					
0	1	1	0	1	0	0	0					
0	1	1	0	1	1	0	0					
0	0	1	1	0	0	0	0					

- Vul nu in kolommen Q_0 t/m Q_1 de binaire waarde in van de hexadecimale getallen die in kolom Q zijn ingeschreven.

- Wat is de invloed van getal B op het resultaat?
- Wat is het verschil tussen getal A en resultaat Q op iedere regel?
- Verklaar ook de aanwezigheid van een punt links van het resultaat op de laatste regel.
- Breek de schakeling NIET af.

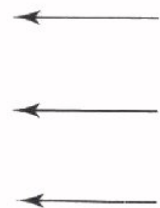
CONCLUSIE

De titel van de opdracht zegt het al: de bits van het getal A zijn elk een bit naar links geschoven ("to shift" betekent schuiven, denk aan shiftregister). Dit noemen we een shift over een bit naar links. Getal B heeft geen invloed op het resultaat, want de ALU kijkt alleen maar naar het getal A; het getal B wordt door de ALU buiten beschouwing gelaten. De punt die plotseling links van het resultaat komt te staan in de laatste regel is op de volgende manier te verklaren. Als we het 4-bits getal 1100 één bit naar links schuiven krijgen we het resultaat 1000 met een overflow (een overstroming) van "1". Het is alsof er een carry-out naar links optreedt.

FUNCTIE SELECTIE

• In de vorige opdrachten hebben we bij C_0 = H drie mogelijke combinaties van de functieselectie-ingangen S_0 t/m S_3 toegepast. We geven hieronder een tabel met alle mogelijke combinaties van S_0 t/m S_3 bij C_0 = H. Achter elke combinatie staat de functie die de ALU daarbij verricht. De besproken functies zijn met een pijl aangeduid.

	M	S_3	S_2	S_1	S_0	Verrichte functie met C_0 = H (geen carry-in)
0	L	L	L	L	L	A
1	L	L	L	L	H	A + B
2	L	L	L	H	L	B + $\overline{B}$
3	L	L	L	H	H	minus 1
4	L	L	H	L	L	A + A.$\overline{B}$
5	L	L	H	L	H	(A + B) plus A.$\overline{B}$
6	L	L	H	H	L	A minus B minus 1
7	L	L	H	H	H	A.$\overline{B}$ minus 1
8	L	H	L	L	L	A plus A.B
9	L	H	L	L	H	A plus B
10	L	H	L	H	L	(A + $\overline{B}$) plus A.B
11	L	H	L	H	H	A.B minus 1
12	L	H	H	L	L	Shift links A
13	L	H	H	L	H	(A + B) plus A
14	L	H	H	H	L	(A + $\overline{B}$) plus A
15	L	H	H	H	H	A minus 1



Betreffende de tabel zijn volgende opmerkingen van belang.

- De controle-ingangen van de ALU zijn M en S_0 t/m S_3. In de tabel hebben we steeds de z.g. mode-ingang M gelijk aan L gehouden. Als we dit doen, zal de ALU bij de diverse combinaties van S_0 t/m S_3 overwegend arithmetische (= REKENKUNDIGE) bewerkingen van twee getallen A en B uitvoeren. Bijvoorbeeld:

```
A          = 0 1 0 1
B          = 0 0 0 1
             ------- plus
A plus B   = 0 1 1 0
```

- Houden we daarentegen steeds M = H aan, dan zal de ALU bij de diverse combinaties van S_0 t/m S_3 LOGISCHE bewerkingen van twee getallen A en B uitvoeren. Bijvoorbeeld:

```
A    = 0 1 0 1
B    = 0 0 0 1
       ------- AND
A.B  = 0 0 0 1
```

  Op de logische bewerkingen komen we later in deze les terug.

- Om verwarring tussen rekenkundige en logische bewerkingen te vermijden, noteren we:

  - in geval van de rekenkundige bewerkingen optellen en aftrekken "plus" en "minus";
  - in geval van de logische bewerkingen AND, OR en EX-OR "•", "+" en "⊕".

- De regels 1, 2 en 3 van de tabel betreffen puur logische functies. De regels 6, 9, 12 en 15 betreffen puur rekenkundige bewerkingen. De regels 5, 7, 8, 10, 11, 13 en 14 betreffen combinaties van rekenkundige en logische bewerkingen.

  Bijvoorbeeld in geval van regel 5 verricht de ALU eerst de logische bewerking A + B en $A.\overline{B}$, om de resultaten daarvan tenslotte rekenkundig op te tellen.

  Ga voor Uzelf na hoe het met elk van de andere regels is gesteld.

- De vermelde verrichtingen gelden als bovendien de carry-in C_0 = H = 0. Is echter C_0 = L = 1, dan wordt bij elk resultaat nog eens 1 opgeteld. Zo zal dan bijvoorbeeld bij regel 11 de logische bewerking A.B worden uitgevoerd.

LOGISCHE FUNCTIES

Tot nu toe hebben we altijd ingang M laag gelaten. Als M laag is, zal de ALU (zoals eerder vermeld) overwegend wiskundige handelingen doen zoals optellen of aftrekken. Het schuiven naar links kan ook als een wiskundige handeling worden beschouwd. Bekijk het volgende voorbeeld:

A = 0 1 0 0 (decimaal 4)

Laten we het getal A naar links schuiven, dan krijgen we:

B = 1 0 0 0 (decimaal 8)

Het schuiven naar links betekent het vermenigvuldigen met 2 en dat is een rekenkundige handeling. Dat de ALU ook logische functies kan verrichten zult U ervaren in de volgende opdracht, waarbij M = H gehouden wordt.

OPDRACHT 8.8: EEN LOGISCHE FUNCTIE

- Zet S_0 = H, S_1 = H, S_2 = L en S_3 = H.
  C_0 mag in deze opdracht zowel L als H zijn; dit kunt U tijdens de opdracht zelf bekijken.
- Verwijder de verbinding tussen C (op de ALU) en de 13-segment indicator.
- Zet de mode-ingang M op H; verbind dus M met P (5V).
- Vul nu proefsgewijze de kolom Q van volgende tabel in.

A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	Q	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0	0	0	0					
1	0	0	0	1	0	0	0					
1	0	0	0	0	1	0	0					
1	1	0	0	0	1	0	1					
1	0	0	1	0	0	1	1					

- De resultaten in kolom Q zijn in hexadecimale vorm weergegeven. Vul in kolommen Q_0 t/m Q_3 de equivalente binaire vorm in.

- Welke is de relatie tussen A, B en Q? Denk aan een fundamentele logische functie.

- Breek de schakeling NIET af.

CONCLUSIE

Indien U, bijvoorbeeld via de controle bus, de ALU opdracht geeft de controle-ingang als volgt in te stellen

$S_0 = H,\ S_1 = H,\ S_2 = L,\ S_3 = H$ en $M = H$,

dan gaat de ALU bit voor bit de AND functie van de getallen A en B maken.

Neem de laatste regel van de tabel als voorbeeld:

$$\left.\begin{array}{l} A_0.B_0 = 1\ .\ 0 = 0 \\ A_1.B_1 = 0\ .\ 0 = 0 \\ A_2.B_2 = 0\ .\ 1 = 0 \\ A_3.B_3 = 1\ .\ 1 = 1 \end{array}\right\} \text{ of ook } \left\{\begin{array}{l} A = A_3A_2A_1A_0 = 1\ 0\ 0\ 1 \\ B = B_3B_2B_1B_0 = 1\ 1\ 0\ 0 \\ \hline \quad A.B = 1\ 0\ 0\ 0 \end{array}\right. \text{ AND}$$

DE OR-FUNCTIE

De ALU kan ook bit voor bit de OR-functie van twee getallen bepalen. Bekijk dit met de volgende programmering:

$S_0 = L,\ S_1 = H,\ S_2 = H,\ S_3 = H$ en $M = H$

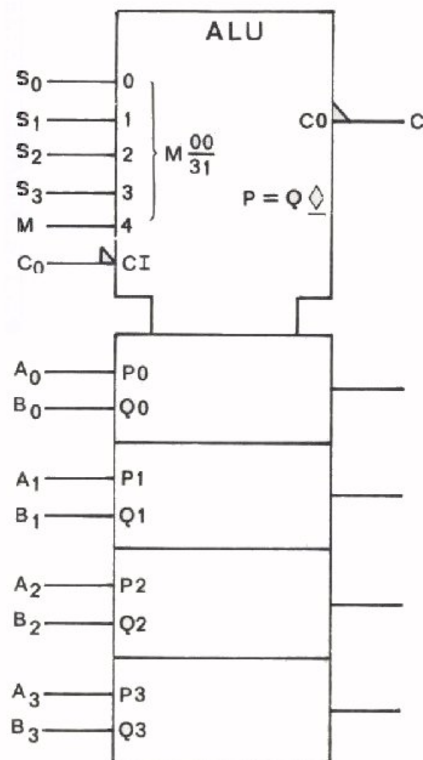
DE LOGISCHE BEWERKINGEN

De mode-ingang bepaalt of de ALU overwegend rekenkundige of logische bewerkingen verricht. In de onderste tabel bekijken we alle logische mogelijkheden bij M = H. De regelnummers sluiten aan op die van de vorige tabel waarbij H = L gold. U heeft reeds ervaring opgedaan met de AND- en de OR-functie. U kunt nog enkele andere uitkiezen en proefsgewijs nagaan.

	M	S_3	S_2	S_1	S_0	Verrichte functie (C_0 = H of C_0 = L		
16	H	L	L	L	L	A		
17	H	L	L	L	H	$\overline{A + B}$	inverterende OR	
18	H	L	L	H	L	$\overline{A} . B$		
19	H	L	L	H	H	0		
20	H	L	H	L	L	$\overline{A . B}$	inverterende AND	
21	H	L	H	L	H	$\overline{B}$		
22	H	L	H	H	L	$A \oplus B$	EX-OR	
23	H	L	H	H	H	$A . \overline{B}$		
24	H	H	L	L	L	$\overline{A} + B$		
25	H	H	L	L	H	$\overline{A \oplus B}$	inverterende EX-OR	
26	H	H	L	H	L	B		
27	H	H	L	H	H	$A . B$	AND	←
28	H	H	H	L	L	1		
29	H	H	H	L	H	$A + \overline{B}$		
30	H	H	H	H	L	$A + B$	OR	←
31	H	H	H	H	H	A		

DE INSTRUCTIECODE

Hieronder ziet U opnieuw het symbool van de ALU en daarnaast de tabellen die nu in één tabel zijn verenigd.



		M	S_3	S_2	S_1	S_0	Verrichte functie als C_0 = H
0	00	L	L	L	L	L	A
1	01	L	L	L	L	H	A + B
2	02	L	L	L	H	L	A + $\overline{B}$
3	03	L	L	L	H	H	minus 1
4	04	L	L	H	L	L	A + A . $\overline{B}$
5	05	L	L	H	L	H	(A + B) plus A . $\overline{B}$
6	06	L	L	H	H	L	A minus B minus 1
7	07	L	L	H	H	H	A . $\overline{B}$ minus 1
8	08	L	H	L	L	L	A plus A . B
9	09	L	H	L	L	H	A plus B
10	0A	L	H	L	H	L	(A + $\overline{B}$) plus A . B
11	0B	L	H	L	H	H	A . B minus 1
12	0C	L	H	H	L	L	shift links A
13	0D	L	H	H	L	H	(A + B) plus A
14	0E	L	H	H	H	L	(A + $\overline{B}$) plus A
15	0F	L	H	H	H	H	A minus 1
16	10	H	L	L	L	L	$\overline{A}$
17	11	H	L	L	L	H	$\overline{A + B}$
18	12	H	L	L	H	L	$\overline{A}$. B
19	13	H	L	L	H	H	0
20	14	H	L	H	L	L	$\overline{A . B}$
21	15	H	L	H	L	H	$\overline{B}$
22	16	H	L	H	H	L	A ⊕ B
23	17	H	L	H	H	H	A . $\overline{B}$
24	18	H	H	L	L	L	$\overline{A}$ + B
25	19	H	H	L	L	H	$\overline{A \oplus B}$
26	1A	H	H	L	H	L	B
27	1B	H	H	L	H	H	A . B
28	1C	H	H	H	L	L	1
29	1D	H	H	H	L	H	A + $\overline{B}$
30	1E	H	H	H	H	L	A + B
31	1F	H	H	H	H	H	A

In de laatste kolom hebben we aangenomen dat C_0 = H. Dit is alléén van belang voor de regels 0 t/m 15. Als namelijk geldt C_0 = L, dan moet in deze regels bij het resultaat 1 worden opgeteld. Voor de overige regels (16 t/m 31) mag C_0 een willekeurig niveau hebben. Naast de regelnummers 0 t/m 31 ziet U hun hexadecimale equivalent (00 t/m 1F).

Zet even voor Uzelf, H = 1 en L = 0 veronderstellend, de niveaus L en H van M, S_3, S_2, S_1 en S_0 om in de waarden 0 en 1. U zult dan merken dat ze, in bovengenoemde volgorde, de binaire uitdrukking van de regelnummers vertegenwoordigen.

Het regelnummer in binaire code legt aldus vast welke signalen men aan de controle-ingangen M, S_3, S_2, S_1 en S_0 moet toevoeren om de ALU de op deze regel vermelde operatie te laten uitvoeren. Het binaire regelnummer noemt men wel de INSTRUCTIECODE. Beschrijft een programmeur instructie-opdrachten dan vermeldt hij de instructiecodes gewoonlijk in hexa-decimale vorm.
Dit heeft als voordelen t.o.v. de binaire vorm, dat het korter is en minder gauw tot vergissingen leidt.

Wilt U nu bijvoorbeeld, dat de ALU de rekenkundige som van A en B maakt (regel 9), dan moet U de ALU de instructie $09_{16} = 01001_2$ geven:

M	S_3	S_2	S_1	S_0		
↓	↓	↓	↓	↓		
0	1	0	0	1	=	09_{16}
= L	H	L	L	H	in geval van H = 1 en L = 0.	

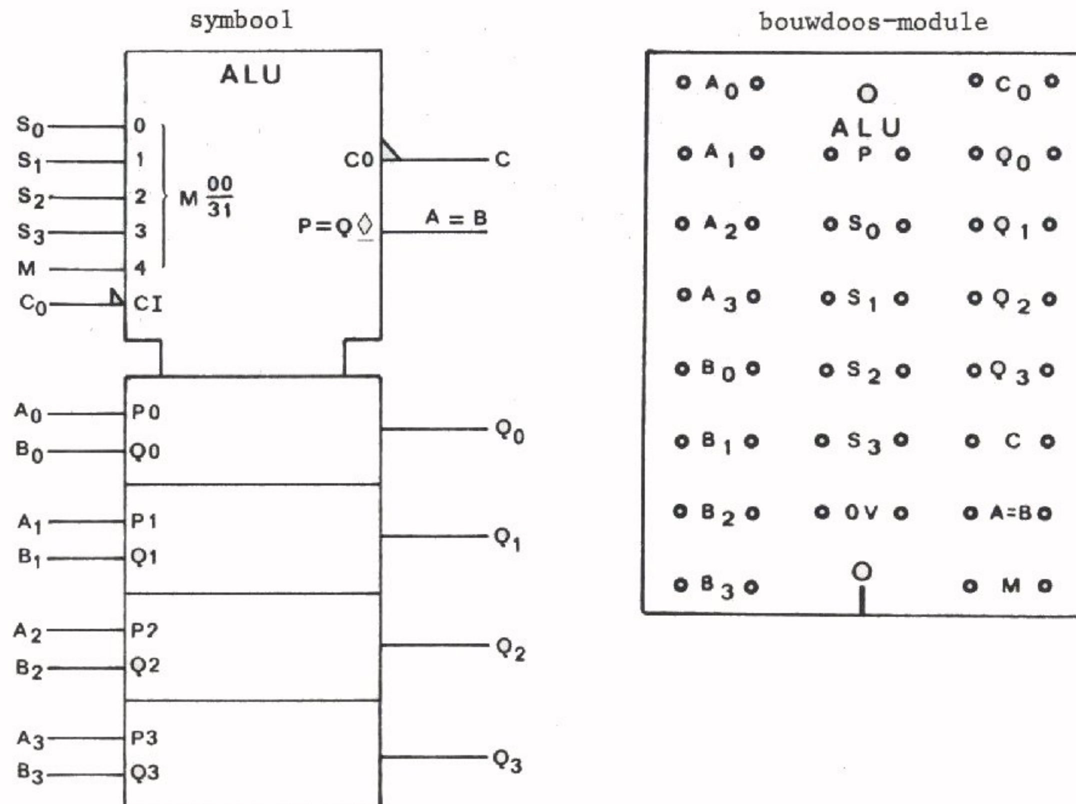
Als U wilt dat de ALU de functie verricht van een inverterende AND-poort, dan moet U hem de instructie 14_{16} geven (controleer dit in de tabel):

M	S_3	S_2	S_1	S_0		
↓	↓	↓	↓	↓		
1	0	1	0	0	=	14_{16}
= H	L	H	L	L	in geval van H = 1 en L = 0.	

In het symbool van de ALU staat de aanduiding M $\frac{00}{31}$. Dit betekent dat via de controle-ingangen (binnen het kader met M0 t/m M4 aangeduid) 32 uiteenlopende instructies kunnen worden gegeven. Een dergelijke aanduiding bent U ook reeds in het symbool van de RAM tegengekomen.

SAMENVATTING

De ALU (= Arithmetic and Logic Unit) is de eenheid binnen een computer die binaire data rekenkundig en/of logisch verwerkt. Besproken is het type 74181, dat in staat is twee 4-bits signalen A en B te verwerken.



INGANGEN zijn er voor:

- het te verwerken signaal $A = A_3A_2A_1A_0$
- het andere te verwerken signaal $B = B_3B_2B_1B_0$
- een carry-in C_0
- de controle- of selectie-signalen S_0, S_1, S_2, S_3 en M

UITGANGEN zijn er voor:

- het verwerkingsresultaat $Q = Q_3\ Q_2\ Q_1\ Q_0$
- een carry-out C
- een vergelijkingsresultaat "A = B"

Behalve de carry-in- en carry-out-aansluitingen, zijn alle aansluitingen hoog actief (zie symbool).

functieselectie-tabel

		M	S_3	S_2	S_1	S_0	Verrichte functie als C_0 = H
0	00	L	L	L	L	L	A
1	01	L	L	L	L	H	A + B
2	02	L	L	L	H	L	$A + \overline{B}$
3	03	L	L	L	H	H	minus 1
4	04	L	L	H	L	L	$A + A . \overline{B}$
5	05	L	L	H	L	H	(A + B) plus $A . \overline{B}$
6	06	L	L	H	H	L	A minus B minus 1
7	07	L	L	H	H	H	$A . \overline{B}$ minus 1
8	08	L	H	L	L	L	A plus A . B
9	09	L	H	L	L	H	A plus B
10	0A	L	H	L	H	L	$(A + \overline{B})$ plus A . B
11	0B	L	H	L	H	H	A . B minus 1
12	0C	L	H	H	L	L	shift links A
13	0D	L	H	H	L	H	(A + B) plus A
14	0E	L	H	H	H	L	$(A + \overline{B})$ plus A
15	0F	L	H	H	H	H	A minus 1
16	10	H	L	L	L	L	$\overline{A}$
17	11	H	L	L	L	H	$\overline{A + B}$
18	12	H	L	L	H	L	$\overline{A} . B$
19	13	H	L	L	H	H	0
20	14	H	L	H	L	L	$\overline{A . B}$
21	15	H	L	H	L	H	$\overline{B}$
22	16	H	L	H	H	L	$A \oplus B$
23	17	H	L	H	H	H	$A . \overline{B}$
24	18	H	H	L	L	L	$\overline{A} + B$
25	19	H	H	L	L	H	$\overline{A \oplus B}$
26	1A	H	H	L	H	L	B
27	1B	H	H	L	H	H	A . B
28	1C	H	H	H	L	L	1
29	1D	H	H	H	L	H	$A + \overline{B}$
30	1E	H	H	H	H	L	A + B
31	1F	H	H	H	H	H	A

De controlesignalen S_0, S_1, S_2, S_3 en M worden vastgelegd in de vorm van binaire getallen M S_3 S_2 S_1 S_0.

Een programmeur noteert deze getallen HEXADECIMAAL voor de eenvoud en om vergissinge te vermijden.

Merk op dat alle controle-ingangen hoog actief zijn, zodat geldt H = 1 en L = 0.

Bij M = L treden overwegend rekenkundige bewerkingen op.

Bij M = H treden overwegend logische bewerkingen op.

Als C_0 = H, dan wordt de vermelde functie verricht;

Als C_0 = L, dan geschiedt "vermelde functie plus 1".

Als REKENKUNDIGE BEWERKINGEN zijn (bij M = H) behandeld:

1. Optellen.

   Daarbij kan zowel een carry-in als een carry-out verwerkt worden door de ALU 74181.

2. Aftrekken.

   Dit geschiedt binnen de ALU m.b.v. de 2-complement methode:

   als de carry-out C = L, dan is de uitkomst positief;

   als de carry-out C = H, dan is de uitkomst negatief.

3. A met B vergelijken.

Daarbij wordt binnen de ALU het "1-complement van B" afgetrokken van "A" (dus C_0 = L):

als C = L en "A = B" = L, dan geldt A > B;

als C = H en "A = B" = L, dan geldt A < B;

als C = H en "A = B" = H, dan geldt A = B.

Als LOGISCHE BEWERKINGEN zijn (bij M = L) behandeld:

1. De logische AND-functie.

2. De logische OR-functie.

